ARMY RESEARCH LABORATORY

Cluster Recognition Algorithms for Battlefield Simulation

Deborah A. Trytten

ARL-CR-288 JANUARY 1996

prepared by

School of Computer Science
University of Oklahoma
Norman, Oklahoma

under contract

DAAL03-91-C-0034

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
January 1996

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Cluster Recognition Algorithms for Battlefield Simulation

5. FUNDING NUMBERS
PE: 6.27.16
PR: 1L162716AH70
DAAL03-91-C-0034

6. AUTHOR(S)
Trytten, D. A.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
School of Computer Science
University of Oklahoma
Norman, Oklahoma

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Human Research & Engineering Directorate
Aberdeen Proving Ground, MD 21005-5425

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
ARL-CR-288

11. SUPPLEMENTARY NOTES
The contracting officer's representative (COR) is Dr. Linda Pierce, U.S. Army Research Laboratory, ATTN: AMSRL-HR-MF, FT Sill, OK 73503 (telephone 405-442-5051).

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The target acquisition fire support model (TAFSM) is a large scale, automated, artillery combat simulation model. This model has been developed over a period of years to include increasingly large and realistic battlefield situations and still retain its efficiency. To process more complicated battlefield scenarios within time constraints, units on the battlefield must be grouped into clusters, which will then be treated as a single entity in further processing. These clusters fall into three categories: circular clusters, linear clusters, and on-line clusters. To perform clustering within the time constraints, two classic clustering algorithms were modified to meet the stringent efficiency constraints (Jain & Dubes, 1988). To detect circular clusters, a template-matching technique was designed. Linear and on-line clusters were found using an algorithm that has roots in the single-link clustering method. These algorithms were implemented and tested on 59 data sets from the TAFSM simulation. The new algorithms ran within a few seconds and created reasonable clusterings.

14. SUBJECT TERMS
clustering
fire support
simulation

15. NUMBER OF PAGES
227

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT

AMS 622716.H700011

Technical Report ARL-CR-288

CLUSTER RECOGNITION ALGORITHMS FOR BATTLEFIELD SIMULATION

Deborah A. Trytten

January 1996

APPROVED: ______________________
ROBIN L. KEESEE
Director, Human Research &
Engineering Directorate

Approved for public release; distribution is unlimited.

U.S. ARMY RESEARCH LABORATORY

Aberdeen Proving Ground, Maryland

ACKNOWLEDGMENTS

This work was supported by the Human Research & Engineering Directorate (Dr. Linda Pierce) of the U.S. Army Research Laboratory under the auspices of the U. S. Army Research Office Scientific Services Program administered by Battelle (Delivery Order 774, Contract No. DAA-91-C-0034).

The expertise, encouragement, and friendship of the personnel working on the TAFSM project were greatly appreciated. Special thanks go to Robin Sexton, Ron Laird, Bill Millspaugh, and Kathy Foster.

CONTENTS

DESCRIPTION OF THE PROBLEM 3

LINEAR AND ON-LINE CLUSTERS 6

Single- and Complete-Link Clustering 7
Two Levels of Clustering 10

CIRCULAR CLUSTERS 17

RESULTS AND CONCLUSIONS 19

REFERENCES 23

APPENDIX

A. Experimental Results 25

DISTRIBUTION LIST 219

FIGURES

1. Three Different Types of Clusters 4
2. A Sample Input to the Clustering Algorithm 4
3. A Sample Output Showing Linear and On-line Clusters 5

TABLES

1. The Parameters Used in the Initial Clustering 13
2. Compatible Cluster Types 14
3. The Parameters Used in the Linear Clustering 16
4. The Parameters Used in the Circular Clusterings 19
5. Combined Execution Time for the Linear and On-line Clustering Algorithms 21
6. Execution Time for the Circular Clustering Algorithm 21
7. The Parameters Used in the Algorithm for Linear and On-line Clustering 22

CLUSTER RECOGNITION ALGORITHMS FOR BATTLEFIELD SIMULATION

DESCRIPTION OF PROBLEM

The target acquisition fire support model (TAFSM) is a large scale, automated, artillery combat simulation model. This model has been developed over several years to include increasingly large and realistic battlefield situations and still retain its efficiency. TAFSM models two competing forces. Battlefield assets such as weapons, communications systems, and detection systems are assigned to each force. Assets are arranged on a flat, two-dimensional battlefield, and a battle is simulated with each force following a prescribed strategy. The battlefield events are all stochastic, so each has a probability of success and failure, and multiple runs are used to analyze a given situation. TAFSM runs in roughly ten times real time, which means that 10 hours of battlefield simulation can be run in 1 hour of computer time. This impressive efficiency is necessary for research results of weapons systems being developed to proceed in a timely fashion.

To process more complicated battlefield scenarios within the necessary time constraints, units on the battlefield must be grouped into clusters, which will then be treated as a single entity in further processing. Units on the battlefield represent the battlefield assets (such as vehicles, weapons, or platoons). This strategy will only improve the system performance if the clustering algorithm is both efficient and reliable. A clustering algorithm that can work like a forward observer, grouping sensor data from objects into perceptually meaningful clusters, is useful in this situation. As an example, objects that are spatially proximate and moving in the same direction should be clustered.

Clusters fall into three categories: circular, linear, and on line. Circular clusters are formed from a group of congregated vehicles. Linear clusters are formed from a convoy headed down a road. On-line clusters are formed as the units spread across the front line of a battlefield. These three types of clusters represent the most strategically important arrangements of assets on the battlefield. Figure 1 shows the three different types of clusters. Note that circular clusters are often stationary and are therefore shown without reference to a direction of movement.

The purpose of the present research is to design and implement clustering algorithms to detect circular, linear, and on-line clusters from sensor data that are modeled in the TAFSM simulation. Input to the clustering algorithms includes the position, velocity, and direction of movement of units on the battlefield. A set of clustering parameters that describes the desired clusters is also given. Output from the algorithm will be a set of clusters of units satisfying the

criterion prescribed by the input parameters. For the clustering algorithms to be successful, it is necessary that they be able to run in just a few seconds, even on a large data set. In addition, the clusters produced must be perceptually meaningful. Figure 2 shows a battlefield with units displayed as dots. The linear clusters produced from this battlefield are shown in Figure 3. The boxes in Figure 3 represent the clusters.

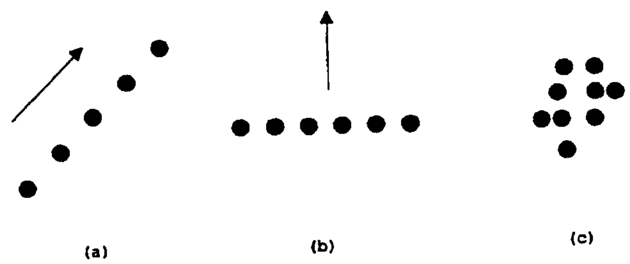

Figure 1. Three different types of clusters.

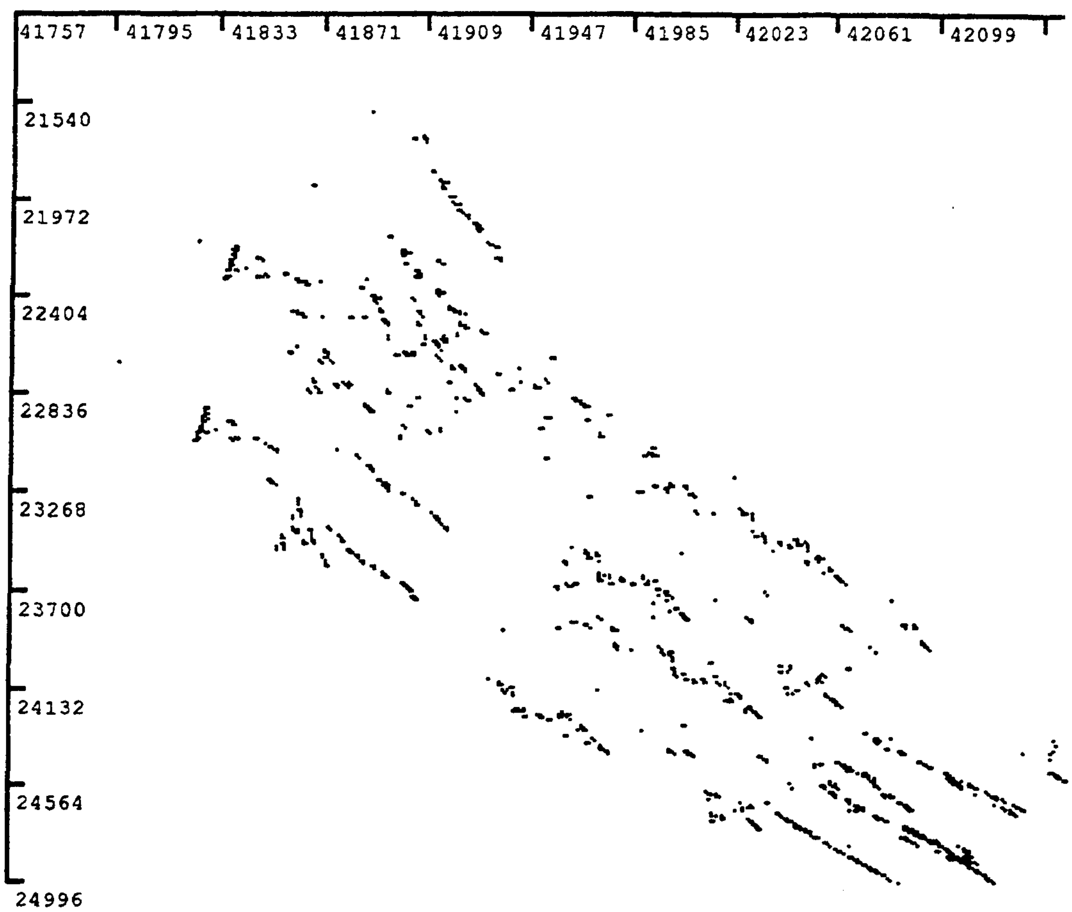

Figure 2. A sample input to the clustering algorithm.

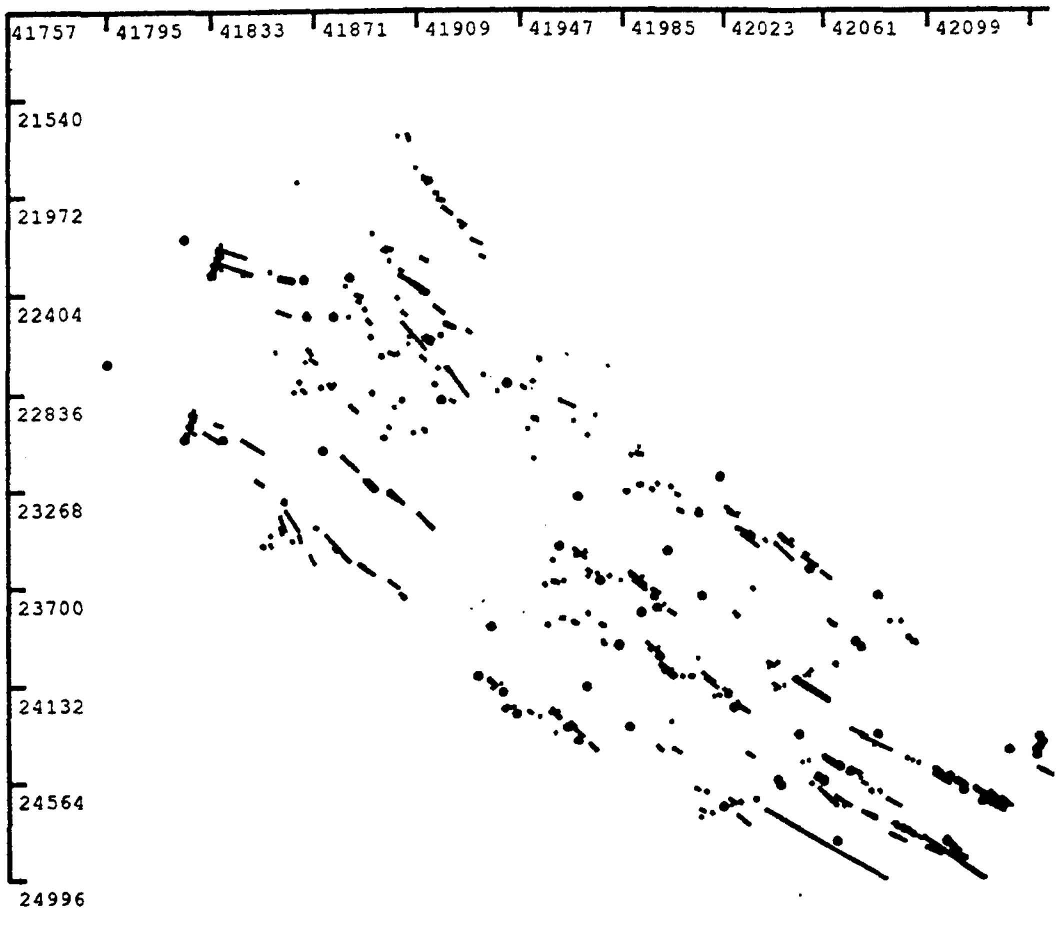

Figure 3. A sample output showing linear and on-line clusters.

Before any clustering algorithm is implemented, it must be established that the data should be clustered. This is necessary because clustering algorithms will generate clusters independent of the existence of clusters in the data. In this application, it is clear that clusters are appropriate since convoys, the front lines of the battlefield, and resupply areas are all examples of clusters that should be anticipated on physical grounds. There are many areas in the sample data sets where the data have very little coherent structure. In most cases, low structural coherence results in low clustering.

The organization of this report is as follows. The next section describes the clustering algorithm designed for linear and on-line clusters. The third section describes the clustering algorithm for circular clusters. The final section summarizes the contributions of this work, draws conclusions, and gives a table of parameters. The appendix shows all 59 data sets,

complete with the initial results, intermediate results, and final results for the linear and on-line clusterings. Results for the circular clustering algorithm are shown for some data sets.

LINEAR AND ON-LINE CLUSTERS

Clustering of linear and on-line clusters is performed by the same algorithm because both types of clusters have the same structure. In both cases, subsets of data that have similar velocities and directions are grouped into linear structures. For a cluster to be either linear or on line, it must look like a rectangle with one pair of sides substantially longer than the other pair of sides. The direction of the longer pair of sides will be referred to as the *cluster direction.* Linear clusters can be distinguished from on-line clusters by the direction of movement of the units, as compared to the cluster direction. Linear clusters move in the cluster direction. On-line clusters move perpendicular to the cluster direction.

Some clustering algorithms are designed specifically for linear structures. An example of one such algorithm is the Hough transform (Hough, 1962). A Hough space is a discretized parameter space. Each bin in the Hough space represents a small range of parameter values. Points in the input image are placed into every bin in the Hough space with consistent parameters. For linear objects, the Hough space is two-dimensional and can be thought of as representing the slope and intercept of a line, although a different parameterization with better properties is commonly used. Since a single point could be on many different lines, the point is placed into many different bins. After all points have been placed into bins, the bin with the maximum number of points is selected as a linear structure. In this way, the set of parameters that represents the maximum number of points in the image is selected as the set of the best line.

The Hough transform is able to select long, straight lines in the presence of many types of noise. A difficulty with the Hough transform is that global structures are detected instead of local ones. For this reason, Hough spaces are inappropriate for clustering in TAFSM since local linearity is what is sought. Hough transforms have been generalized to arbitrary parametric shapes (Ballard, 1981). As an example, a circle has three parameters: the two coordinates of the location of the center of the circle and the radius. A Hough space could be used to detect circular clusters, but the size of the parameter space is large. A survey of Hough transform techniques, including both the benefits and difficulties, is available in a report by Illingworth and Kittler (1988). Difficulties with the Hough space, particularly in the presence of noise have been discussed (Grimson & Huttenlocher, 1988).

More general clustering algorithms, which are often based on graph theory, fall into two categories: partitional algorithms and hierarchical algorithms. A partitional clustering algorithm is one that starts with all the points grouped into a single cluster and then repetitively divides cluster(s). A partitional clustering algorithm was proposed by Zahn (1971). Zahn suggests that the minimum spanning tree (MST) of the data points be created. Any edge in the MST that is longer than a threshold parameter is then removed, leaving connected components of the tree as clusters. A difficulty of this method is developing a method for selecting break points that will respect the linearity of the desired clusters without examining a multiplicity of subsets around every point.

The second type of general clustering algorithm is the hierarchical. Hierarchical clustering algorithms begin with all the points representing one-element clusters. The clusters are then hierarchically assembled into larger clusters, based on some joining criterion. One such clustering algorithm was proposed by Tuceryan (1986). In this method, only clusters that are neighbors according to the Voronoi tesselation definition of neighborhoods are considered for joining. A difficulty with this algorithm is that the Voronoi tesselation, which is $O(n \log n)$, must be found for the n points.

Another set of hierarchical clustering algorithms tries to join small clusters, based on a distance metric. Clusters are joined by ascending distance metric. A variety of metrics has been used, including Euclidean and Manhattan metrics. Both the single-link and complete-link algorithms fall into this classification. These methods form the basis for the hierarchical algorithm designed for the linear and on-line clusters and are further discussed in the next subsection.

This section is divided into two subsections. The first discusses the classic single-link and complete-link clustering algorithms and why they are inadequate for this task. The second describes a hierarchical clustering algorithm derived from the single-link clustering algorithm, which performs both linear and on-line clustering.

Single- and Complete-Link Clustering

Single-link and complete-link clustering algorithms are both based on the following idea: clusters that are closer together should be grouped first. Both the single-link and complete-link clustering algorithms are special cases of a general set of algorithms called sequential,

agglomerative, hierarchical, nonoverlapping (SAHN) algorithms. The general framework for a SAHN algorithm consists of the following steps (Jain & Dubes, 1988):

1. Begin with the disjoint clustering having level $L(0) = 0$ and sequence number $m = 0$.

2. Find the least dissimilar pair of clusters in the current clustering, for example, pair {(r), (s)} according to some distance metric d.

$$d[(r),(s)]=\min\{d[(i),(j)\}]$$

in which the minimum is taken over all pairs of clusters in the current clustering.

3. Increment the sequence number m by 1. Merge clusters (r) and (s) into a single cluster to form the next clustering m. Set the level of the clustering to

$$L(m)=d[(r),(s)]$$

4. Update the proximity matrix D by deleting the rows and columns corresponding to clusters (r) and (s) and adding a row and column corresponding to the newly formed cluster. The proximity between the new cluster denoted (r,s) and old cluster (k) is defined as follows.

For the single-link method,

$$d[(k),(r,s)] = \max\{d[(k),(r)],d[(k),(s)]\}$$

For the complete-link method,

$$d[(k),(r,s)] = \max\{d[(k),(r)],d[(k),(s)]\}$$

5. If all objects are in one cluster, stop. Otherwise, go to Step 2.

The distinction between the complete-link algorithm and the single-link algorithm is that the complete-link algorithm is conservative. Clusters will not be joined until all points in both clusters are proximate. The single-link algorithm is more daring, joining clusters when only a pair of points within the clusters becomes close. Complete-link clusters are joined when the distances between all points in both clusters are minimal among pairs of clusters. In graph theoretic notation, this means that all the edges in a complete subgraph of the pair of cluster points must be of minimal distance among the pairs of clusters. Single-link clusters are joined when the distance between a single pair of points in both clusters is minimal. This means that one of the

edges in the subgraph of cluster points is of minimal distance among pairs of clusters. Since we are not looking for clusters that are compact in the linear and on-line case, complete-link clustering is inappropriate in this application. The shortcomings of complete-link clustering for the circular case are discussed in the section entitled Circular Clusters. Other SAHN algorithms and the variety of distance metrics are given in Jain and Dubes (1988).

There are a number of problems using the single-link algorithm directly for clustering in TAFSM. The first problem is that it does not explicitly allow for considerations other than distance between points to be used in determining whether clusters should be joined. For example, data points in a cluster must have a velocity and direction that are within a given tolerance. This feature could be added to the clustering algorithm with little difficulty.

A second problem is that the single-link method creates a complete hierarchy of clusterings, stopping only when all points in the data set have been joined into a single cluster. After this cluster hierarchy is created, it can be cut at a variety of thresholds to determine the clusters. One method for cutting is to look for a large jump in distance values *(d)* between levels. A large jump in distance value means that a cluster remained unchanged for a long time. This is one indication that the cluster might be perceptually significant. Being able to select different thresholds from a single clustering is important in many clustering applications because there is no physical basis for parameter selection. In TAFSM, there is a physical basis since the parameters of the weapon systems being tested are the variables being examined during the simulation runs. Therefore, the hierarchical clustering can be stopped when these known parameters are exceeded. This greatly improves the efficiency of clustering. Single-link clustering also requires that distances be calculated between all pairs of points initially. This is very inefficient when one is handling many thousands of points, most of which are not proximate.

A final shortcoming with single-link clustering is that the usual distance metrics, Euclidean and Manhattan, are not adequate in this application. Between linear clusters, variations in distance along the cluster direction are expected. Variations in distance perpendicular to the cluster direction should not be treated similarly.

Unlike the difficulties with the other clustering methods discussed, all these shortcomings in the single-link clustering algorithm can be overcome to produce an algorithm that is better tuned to the TAFSM simulation. Two algorithms derived from the single-link algorithm are discussed in the next two subsections.

Two Levels of Clustering

Defining a distance metric to use for joining clusters is difficult to do in one step. For clusters with one or two points, it is difficult to tell whether the cluster is linear or on line. Once a few more points in the cluster are available, this distinction can be easily made. Once a cluster is known to be linear, for example, the parameters that are relevant to a linear cluster can be used to better tune further clustering.

This observation motivates a two-level clustering algorithm. In the first clustering, any data points that are sufficiently close using a Euclidean distance metric and share a common velocity and direction of movement can be joined in ascending order of distance. This is done without any reference to whether the clusters are linear or on line. Once sufficiently large clusters have been created, they can be classified by type. While some clusters will still not have enough data to classify, the vast majority of the clusters can be organized into linear and on-line clusters. These low-level linear and on-line clusters can be clustered as two separate clustering problems. The distance metric used is identical, but the specific parameter values may vary since distances that would exist in a convoy traveling toward the front might differ from distances between adjacent vehicles on the front lines.

It is necessary to create clusters in ascending order of distance rather than just grouping any points of distance less than a threshold. When clusters are being formed, the velocity and direction vectors need to be compatible. If points that are farther away are joined with a cluster instead of the points that were closer, the velocity and direction may be corrupted.

The initial clustering algorithm is discussed next, including the method for classifying clusters as either linear or on line. This is followed by a discussion of the algorithm that clusters the clusters produced by the first clustering algorithm.

Initial Clustering

The initial clusters are created using a modification of the single-link algorithm. The general methodology of joining data points or clusters that have smaller separations first, which was used in the single-link algorithm, will be used. The Euclidean distance metric will be used, since the initial clustering is done before clusters of sufficient size to permit classification have been created.

1. For each pair of points i and j,

if the distance between i and j is less than a threshold value, add (i,j) to an ordered list of pairs P.

2. Sort list P into ascending order, using the Euclidean distance for the comparisons.

3. For each pair (i,j) in the sort list D,

(a) Check to see if points i and j are already in the same cluster. If they are, then go to the next pair of points.

(b) Check to see if the cluster that contains i and the cluster that contains j have velocities and directions that are within a threshold value. If they do, then join these clusters into a single cluster.

4. Create statistics that describe each cluster.

A number of methods were used to improve the efficiency of this algorithm. Instead of calculating distances between all pairs of points i and j, we placed the points in discretized bins. The size of the discretized bins is determined by the maximum cluster size desired. Based on previous experimentation with clustering, Ron Laird contributed the idea that only points in the same or eight connected neighboring bins have distances calculated, since only points within these adjoining bins could be close enough to allow clustering. In the typical data set, this saves an enormous amount of calculation. The size of the bins must be larger than the radius of the small clusters.

The bins are searched in raster order. If the current bin is at location (a,b), (when a and b are integer indices for the bin array), then only comparisons to bins at (a,b), $(a+1,b)$, $(a,b+1)$, $(a+1,b+1)$, and if $b-1>0$, bin $(a+1,b-1)$, need to be made. Points that share a bin are only included in the cluster list if the index of the first point is smaller than the index of the second. This scheme prevents any recalculation of distances and the duplication of points in the list P.

The statistics that are used to describe each cluster are the center, a list of elements, and the eigenvalues and eigenvectors. The eigenvalues and eigenvectors describe the distribution of data in the cluster. The eigenvalues are ordered with the largest eigenvalue and its corresponding eigenvector being labeled as the first, and the other eigenvalue-eigenvector pair being labeled as the second. If the eigenvalues are approximately equal, the cluster could be

circular. Circular clusters are sought by a different algorithm. If one eigenvalue is much larger than the other, the cluster is elongated. The eigenvector that matches the larger eigenvalue is in the direction of maximum variation in the cluster. The second eigenvector is perpendicular to the first. The length and width of the cluster are calculated by finding the points that are farthest away from the center in the direction of the first and second eigenvectors, respectively.

To determine whether a cluster is either on line or linear, the ratio of the first eigenvalue to the second eigenvalue is compared to a threshold. If the ratio of the eigenvalues is larger than the threshold, then the cluster can be labeled as being either on line or linear. If the ratio of the eigenvalues is smaller than the threshold, then the cluster is labeled as being of unknown type. To differentiate linear and on-line clusters, the direction of movement must be compared to the cluster direction. The direction of movement is stored as an angle θ. The cluster direction is represented by the first eigenvector. If the absolute value of the dot product of the first eigenvector and the movement direction vector $(\cos \theta, \sin \theta)$ is bigger than the absolute value of the dot product of the second eigenvector and the same movement direction vector, then the cluster has more variation in the direction parallel to the direction of movement than in the direction perpendicular to the direction of movement. This means that the cluster is moving in the direction of the length of the cluster, that is, it is a linear cluster. If the absolute value of the dot product of the second eigenvector with the movement direction vector is larger than the absolute value of the dot product of the first eigenvector with the movement direction vector, then the movement is in a direction perpendicular to the length of the cluster. This means that the cluster is on line.

Any cluster that cannot be classified is included on a list of unknowns. Unknown clusters, some of which contain only a single point, are allowed to cluster with any linear or unknown cluster. Clusters are not permitted to form hierarchically between clusters labeled as linear and on line. The vast majority of the clusters in the data sets were linear clusters, with only a few on-line clusters.

The appendix shows the calculation of intermediate clusters for all 59 data sets on the page following the initial data. Although it is difficult to detect in the large data sets, only the boxes describing the eigenvectors and eigenvalues are displayed in the initial clusterings; the data points are not displayed. Unknown clusters are displayed with a circle. Linear clusters are denoted with a box made of solid lines. The few on-line clusters that were found are denoted by a dashed box. The boxes in both the linear and on-line case are graphical representations of the

eigenvectors and the extent of the cluster. The parameters used in this algorithm, along with the default values, are given in Table 1.

Table 1

The Parameters Used in the Initial Clustering

	Default value
Threshold on maximum distance for D	8 units
Threshold on velocity differences	1.0 unit
Threshold on directional differences	0.5 radian
Threshold on ratio of eigenvalues	4

Extensive experimentation with parameter values was done, and the results are not sensitive to small changes in parameter values. The most important parameter is the distance threshold. If there is difficulty determining an optimal value, it is best to set this parameter at the small end of the acceptable range since the second level of clustering will be able to join fractured clusters but not subdivide existing clusters. A symptom of the distance threshold being too small is that far too many clusters are labeled as unknown. The data sets given did not have any variation in velocity, so the velocity threshold was set arbitrarily. A much smaller directional difference threshold would have sufficed since the directions of the clusters given in the data had very little variation.

Clustering the Initial Clusters

The algorithm for clustering the initial clusters into more perceptually significant groupings is given here. It is similar to the initial clustering algorithm. This distance metric is not Euclidean in this case. It is described after the algorithm.

1. For each pair of clusters of compatible types i and j,

   if the distance between i and j is less than a threshold value, add (i,j) to a list of ordered pairs P. The distance metric is explained:

2. Sort list P into ascending order of the distance.

3. For each pair (i,j) in the sorted list P,

(a) Check to see if clusters i and j are already joined. If they are, then go to the next pair of clusters in list P.

(b) Check to see if the cluster that contains i and the cluster that contains j have velocities and directions that are within a threshold value. If they do, then tentatively join clusters i and j into a single cluster.

(c) Create statistics that describe the new cluster. The new cluster must be within tolerances before replacing clusters i and j with the new cluster.

The compatible cluster types are described in Table 2. Clusters of unknown type are not permitted to join to other clusters of unknown type. This is necessary to prevent single point clusters at large distances from coming together.

Table 2

Compatible Cluster Types

Linear	Linear
On line	On line
Linear	Unknown
On line	Unknown

The distance metric used to evaluate whether a pair of clusters should be joined compares the distance between the cluster centers in the direction of the eigenvectors. Separate thresholds are used for each eigenvector. For both linear and on-line clusterings, much more tolerance is given in the direction parallel to the first eigenvector than in the second. In addition, the length of the clusters, but not the width, was subtracted from the distance in the direction of the first eigenvector. This is necessary since the cluster centers between two linear clusters joined end to end could be far apart, even if the separation between the end points were small.

A distance metric in which the distances were combined using variable weights was also used during experimentation with this algorithm. The difficulty with this second metric

was that variation in the distance along the second eigenvector would limit the tolerance to variation along the direction of the first eigenvector. This meant that some small gaps would not be filled if the clusters had even a small difference in centers along the direction of the second eigenvector. Reducing the penalty for variation in the direction of the second eigenvector produced wide, nonlinear clusters since two linear clusters that were running in parallel directions could then join.

As in the initial clustering algorithm, cluster end points are again put into discretized bins to avoid having to calculate the distance between all pairs of end points. Cluster end points were determined to be the points farthest from the cluster center in the direction of the first eigenvector for both the linear and on-line clusters. The bins that must be compared were previously described.

When clusters are joined, two additional parameters are examined before the cluster is permanently accepted. First, the width of the cluster is compared to two times the maximum possible distance parallel to the second eigenvector. This keeps linear clusters that are proximate from joining into a single long, wide cluster.

A second test is performed to assure that joined clusters are good. An *accidental alignment* occurs when two small clusters that are spatially separated are joined because they are part of the same line. If there are intermediate points, this joining is not only acceptable but desirable. The worst case here is when a cluster with only three points is the maximum threshold distance parallel to the first eigenvector from a second cluster with only three points. A density threshold comparison is done to determine if there is a sufficient number of points per unit length in the new linear cluster. This eliminates problems with accidental alignments.

The worst case computational complexity for the algorithm is $O(n^2)$ in which n is the number of data points. In this worst case, all n points would be clustered into a single discretized bin so that all the paired distances would have to be calculated. This case is very unlikely to happen in practice. Average case computational complexity was not calculated since a meaningful statistical description of the distribution of the data is not known.

The parameters were set experimentally. Tuning the parameters will take some experimentation when a new modality of data, such as that coming from a new sensor, is presented. The general rules are as follow. If more clustering is desired, all the thresholds except the density threshold should be raised. The density threshold should be lowered. If clusters that

are too big are being created, then all thresholds should be lowered, except for the density threshold which should be raised. The X-windows user interface provided permits a user to enlarge a particular area to see the individual points that comprise a cluster. This can provide valuable insight, particularly with the large data sets in which individual points are difficult or impossible to see at a resolution that will permit viewing the entire field of view.

Jumping immediately to an extreme case of the parameters individually will often give a quick impression of which parameter is preventing or forcing an inappropriate break or join. As an example, if it is suspected that the directional difference may be too low, the threshold can be set to 360°. This removes any possible effect of directions. If an inappropriate gap is not filled with this threshold, then clearly the clustering is being limited by another parameter or a more complex interaction between parameters and no further adjustment of this parameter alone can improve the clustering.

The number of examples of on-line groupings was not sufficient to determine an independent set of parameters. As a result, the values that are given in Table 3 should be treated with suspicion. The software is written in such a way that the parameters for linear and on-line clustering are completely independent.

Table 3

The Parameters Used in the Linear Clustering

Parameter	Default value
Threshold on distance parallel to the first eigenvector	100 units
Threshold on distance parallel to the second eigenvector	8 units
Threshold on velocity differences	1.0 unit
Threshold on directional differences	0.5 radian
Threshold on density	.05

Note. The on-line clustering also used these parameters.

A visual examination of the results in the appendix shows that the linear and on-line clustering algorithm is performing well. There are cases when the results are not perfect, for example, when a relatively small gap between two apparently linear clusters has not been bridged. Investigation of these cases shows that the cluster centers are displaced enough in the direction perpendicular to the first eigenvector that the clusters cannot join. This investigation was conducted by examining the clusters at higher resolution with respect to the data. There are no known cases when this was caused by a directional or velocity difference. The data within the test set have very little variation in the directional and velocity components. There are also some cases when small clusters that are relatively far apart have been joined. If this is deemed to be undesirable, a higher density threshold will keep these clusters from joining.

CIRCULAR CLUSTERS

The algorithm for finding circular clusters is considerably more simple than the linear and on-line algorithm. Since circles are rotationally invariant, a simple template-matching operation can find circular clusters. Suppose that the data set is represented as an image. Let the pixels in the image have a value equal to the number of units that are located at a particular point. Template matching takes a small array, called a mask, and convolves this mask with the image of the data set. The pixel with the highest convolution value is the center of the best circular cluster. The only parameter used is the radius of the mask.

The mask for this convolution operation is binary. A 0 position in the mask indicates that the point corresponding to this mask is outside the region of interest. A 1 position in the mask indicates that the point corresponding to this mask point is inside the region of interest. Convolution is performed by setting the mask on top of the image at each position, multiplying corresponding mask and pixel values and adding the result. The mask is essentially providing an indicator function that determines whether the pixels in adjacent bins should be counted as contributing to a cluster centered at this point in the image. These convolution values are stored, and the maximum value is declared to be the best circular cluster. In the current implementation, a list of the best k clusters, in which k is a parameter, is kept. To calculate the best k clusters, points that are used in the first cluster must be removed from the bins before calculations are performed for the second cluster. If this is not done, points may be in more than one cluster.

The only difficulty with template matching for this application is the time complexity. If the data set has n points, and the size of the data image is m x m, then there are $O(m^2)$ operations. Note that this is the best case, the worst case, and the average case time complexity.

If specialized convolution hardware were available on the system, this algorithm could be run in its original form very quickly. Since this hardware is not available, a more efficient version of the algorithm is sought.

To improve the efficiency of the algorithm, the sparcity of the data set is used to advantage. Instead of having a separate accumulator array for the convolved values, data points will contribute to bins that are within the region of interest on the mask centered at the data point. Accumulators will be stored in a sparse array, so that only bins that have been incremented will need to be searched for maximal counts. With this change, the time complexity of the algorithm is $O(m^2)$ in the worst case, in which every single bin in the entire image plane was used (this is possible only if there is a sufficient number of points, which is not typically the case). In the best case, all n points coincide, and the algorithm is $O(n)$. Notice that the computational complexity of this algorithm is better than that of the linear and on-line clustering. This explains why this algorithm was selected instead of modifying the on-line and linear algorithm previously described.

For a large cluster radius, the execution time of this algorithm as presently described can be too slow. For example, if the mask represents a circle of diameter 20, the approximately 314 bins will be incremented in the neighborhood of the point. Using a sparse matrix avoids searching thousands of empty bins, but there is a penalty in terms of overhead. To improve the efficiency further, a mask of the same shape but with lower resolution was used. By lowering the resolution by a factor of 4, for example, only 13 bins need to be incremented for each point. The price for this lower resolution is that the exact location of the best cluster at the higher resolution is not found. To restate this, the best cluster center need not have its center at the center of the best bin.

If a more accurate position is needed, as might be true in the case of targeting artillery, a high resolution search of the area immediately surrounding good bins could be done. This will be much more efficient, since many points will have been removed from consideration before the high resolution search begins. This improvement was not implemented.

The parameters used in the circular clustering are described in Table 4. The quotient of the diameter of the mask with the resolution needs to be an odd number. This is because circles are represented better in odd size masks than in even size masks. All the parameters in this table can be directly found from application considerations. For example, if artillery that will destroy anything in a 50-unit diameter is to be fired, then the cluster diameter should clearly be 50. If

only one round can be fired, then only one cluster needs to be kept. The minimum number of clusters desired should be requested since some recalculation must occur to obtain further clusters. Two resolution values that meet the constraints for a 50-unit diameter are 2 and 10 units per bin. Using a resolution of 2, or a mask size of 25, would produce more accurate target recognition. The resolution of 10, which corresponds to a mask size of 5, would run much more quickly.

Table 4

The Parameters Used in the Circular Clusterings

Number of good clusters to keep	10
Diameter of the mask	20
Resolution of the mask	4

If the diameter of the area that the artillery is to impact were stochastic, the mask could use values other than unity and zero. Points near the center of the mask would have higher weights, and points away from the center would have lower weights. This idea was not implemented.

In the appendix, some of the data sets have circular clusterings calculated. Although the circular clustering algorithm was run on all 59 data sets, results are reported only for the small sets. In the large data sets, it is impossible to determine if the algorithm is running correctly by inspection. This is particularly true since multiple units may be positioned at a single pixel. Since the images are not useful for evaluating the success of this algorithm, they are not included in this report.

RESULTS AND CONCLUSIONS

This report describes two algorithms for clustering data. The first algorithm can produce both linear and on-line clusters by grouping data points together into small clusters. These small clusters are then classified and grouped into larger clusters. An advantage of the two-level scheme is that a "quick and dirty" initial grouping limits the number of clusters that must be considered for a more rigorous hierarchical grouping. The second algorithm describes a clustering

method for circular clusters. The method produces results that are identical to using a circular template mask but offers improved efficiency.

Both methods have been run on all data sets provided. The clustering results are complete and reasonable. Evaluating a clustering algorithm in objective terms is always problematic since there is no single objective measure of what is a good clustering. Methods such as determining the amount of clustering on random data are not applicable in this domain since it is known that the data are not random. Subjective visual inspection of the processing results, as shown in the appendix, shows that the algorithms are both working well.

Variation of parameters permits different clustering policies to be enforced. A conservative policy would favor small thresholds. This would be appropriate in a case when an extraneous clustering is potentially problematic. If it is more important to improve the efficiency and encourage bigger clusters, the parameters can be adjusted to this policy. The policy enforced in the reported results is not conservative.

Table 5 presents some examples of running times for the linear and on-line clustering algorithms on a SparcStation LX workstation which runs at 59.1 MIPS. Other relevant performance parameters are 4.6 MFLOPS, 26.4 SPECint92, and 21.0 SPECfp92. Table 6 gives the run time on the same workstation with the same performance parameters for the circular clustering algorithm. Before these times were calculated, the graphical user interfaces (GUIs) were disabled so that only the time to read the data and find clusters was recorded. Since data are being read from hard disk, the times provided may be larger than those that will be achieved when the data reside in memory. Although it is impossible to tell what the exact execution speed of the algorithms will be when they are integrated into TAFSM on different hardware, they are currently running quickly. The times for the circular clustering algorithm reflect finding the ten best clusters. There is a time savings if only one good cluster is sought.

Table 7 gives a list of the parameters used to cluster all the sample data sets. A wide range of parameter values could be used with only small differences in outcome. These parameter values can be fine tuned using the X-windows graphical user interface and command line arguments.

Table 5

Combined Execution Time for the Linear and On-line Clustering Algorithms

Data set	No. of points	Time in seconds
0002	46	--
0013	1079	2
0030	1253	2
0058	1977	4
0020	2179	5
0022	2589	6

Note. Times indicated by a (-) in the table are less than 1 second. Time reported is the elapsed time on the clock, not the actual central processing unit (CPU) time used, which is too small to measure for most of the clusterings.

Table 6

Execution Time for the Circular Clustering Algorithm

Data set	No. of points	Time in seconds
0002	46	--
0013	1079	2
0030	1253	2
0058	1977	3
0020	2179	3
0022	2589	4

Note. Times indicated by a (-) in the table are less than 1 second. Time reported is the elapsed time on the clock, not the actual CPU time used, which is too small to measure for most of the clusterings.

Table 7

The Parameters Used in the Algorithm for Linear and On-line Clustering

Initial linear and on-line clustering parameters	
Threshold on maximum distance for *D*	8 units
Threshold on velocity differences	1.0 unit
Threshold on directional differences	0.5 radian
Threshold on ratio of eigenvalues	4
Second level linear and on-line clustering parameters	
Threshold on distance parallel to the first eigenvector	100 units
Threshold on distance parallel to the second eigenvector	8 units
Threshold on velocity differences	1.0 unit
Threshold on directional differences	0.5 radian
Threshold on density	.05
Circular clustering parameters	
Number of good clusters to keep	10
Diameter of the mask	20
Resolution of the mask	4

REFERENCES

Ballard, D.H. (1981). Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition, 13, 111-122.

Grimson, W.E.L., & Huttenlocher, D.P. (1988). On the sensitivity of the Hough transform for object recognition. Proceedings of the Second International Conference on Computer Vision, Miami, FL, 700-706.

Hough, P.V.C. (1962). Method and means for recognizing complex patterns. U.S. Patent No. 3069654.

Illingworth, J., & Kittler, J. (1988). A survey of the Hough transform. Computer Vision, Graphics, and Image Processing, 44, 87-116.

Jain, A.K., & Dubes, R.C. (1988). Algorithms for clustering data. Prentice-Hall.

Tuceryan, M. (1986). Extraction of perceptual structure in dot patterns. Unpublished doctoral dissertation, University of Illinois, Urbana-Champaign.

Zahn, C.T. (1971). Graph-theoretical methods for detecting and describing Gestalt clusters. IEEE Transactions on Computers, C-20, 1.

APPENDIX A

EXPERIMENTAL RESULTS

EXPERIMENTAL RESULTS

This appendix contains printouts of the initial data, intermediate results, and final clusterings for all 59 data sets supplied for the linear and on-line clustering methodologies. Following some data sets are the circular clustering results.

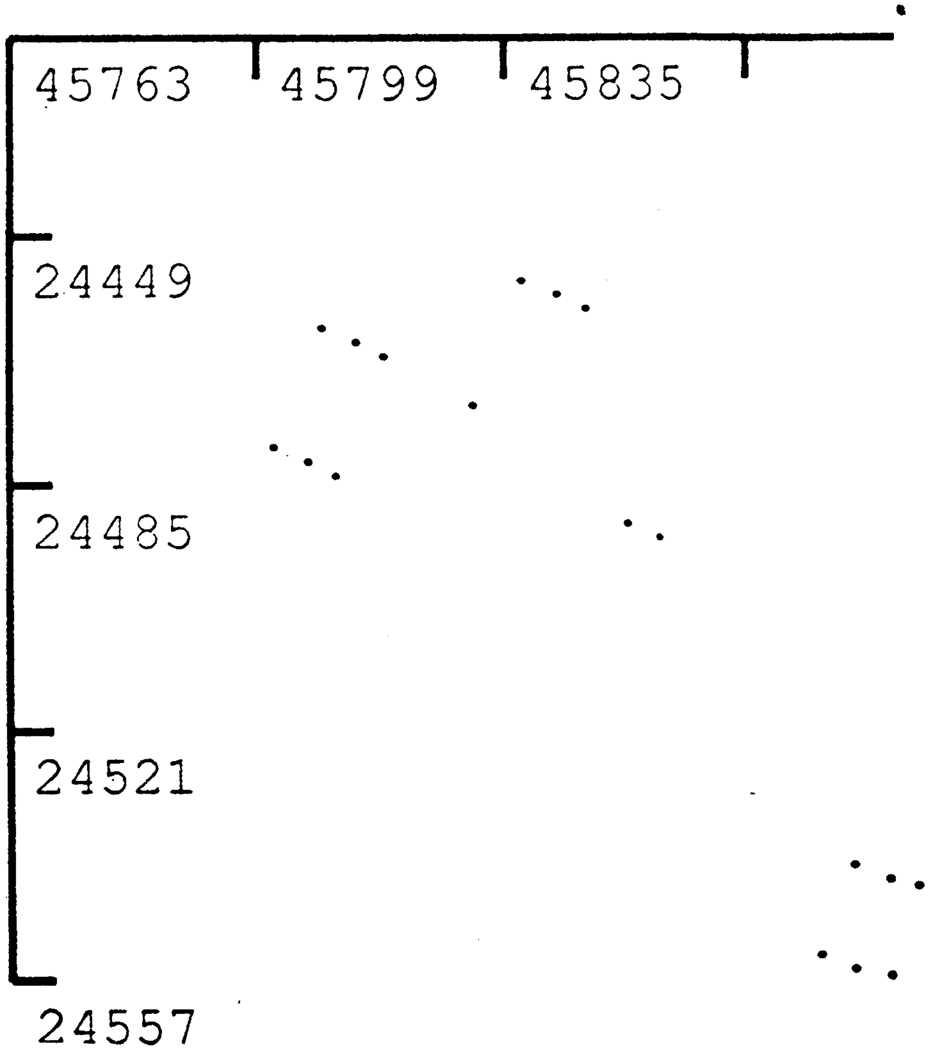

Figure A-1. Input data from data set 1.

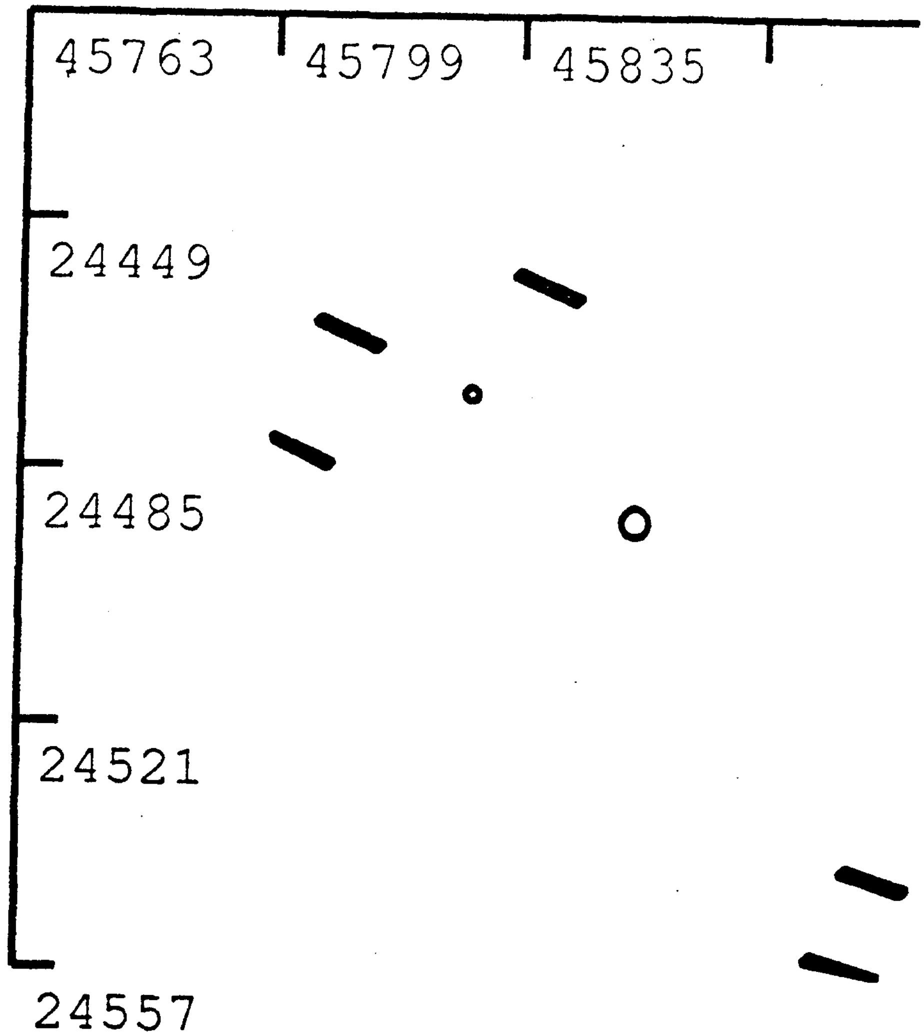

Figure A-2. Initial clusters from data set 1.

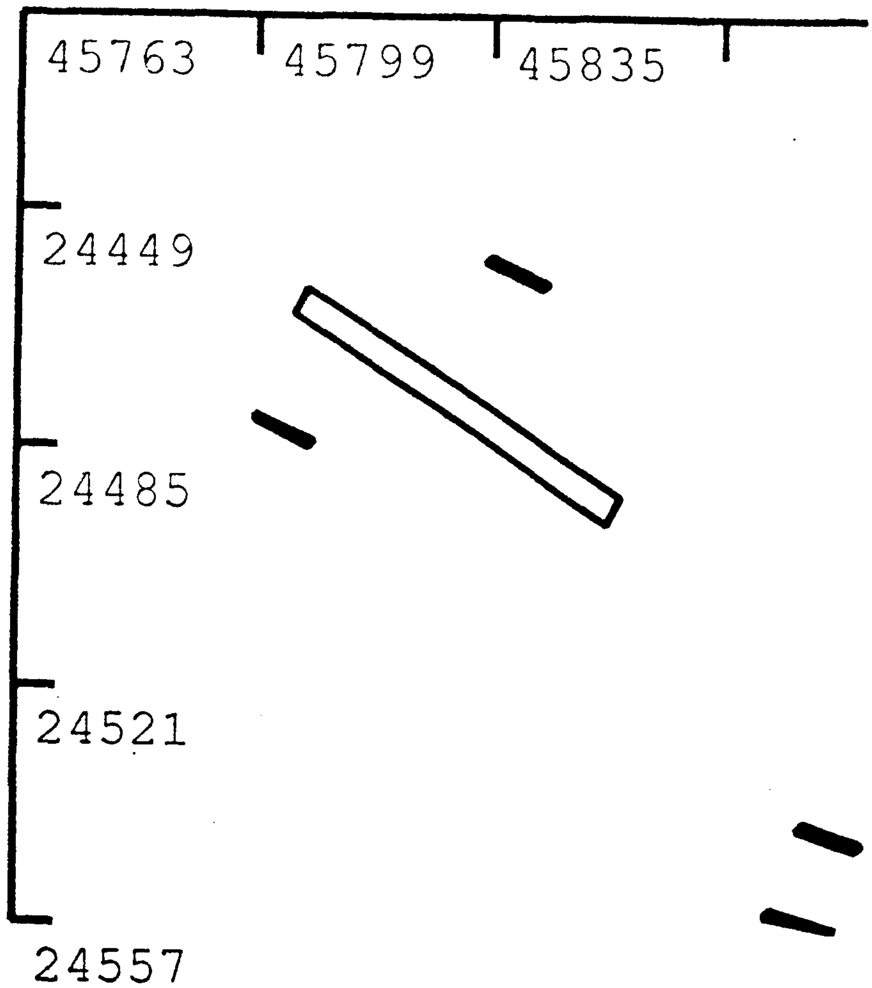

Figure A-3. Linear and online clusters from data set 1.

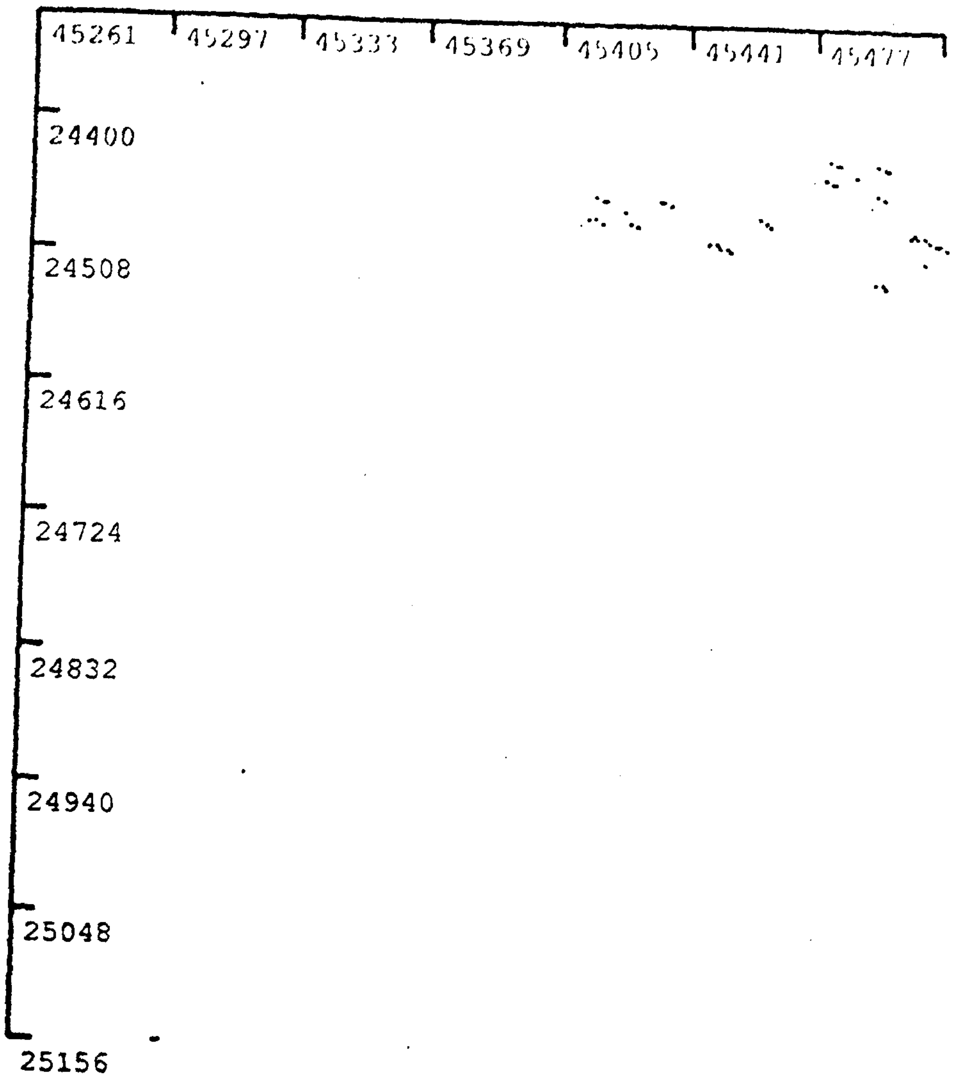

Figure A-4. Input data from data set 2.

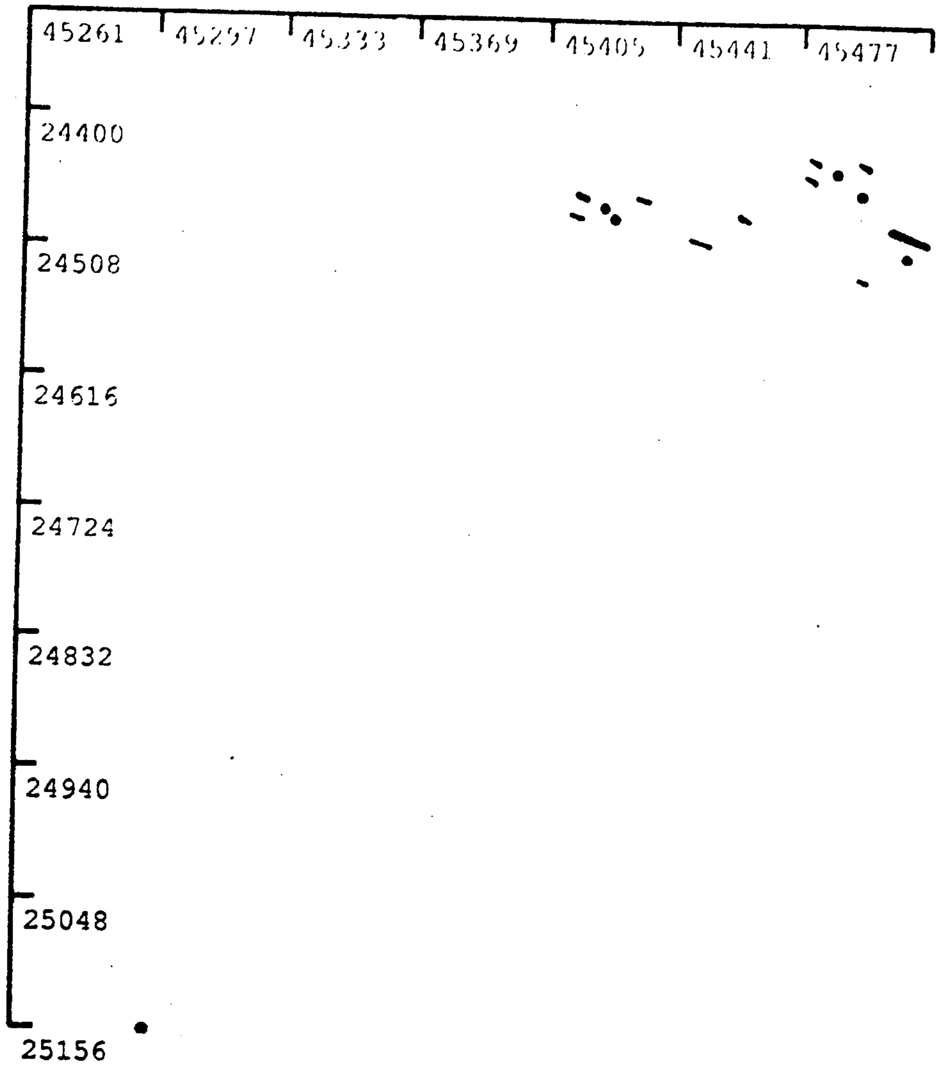

Figure A-5. Initial clusters from data set 2.

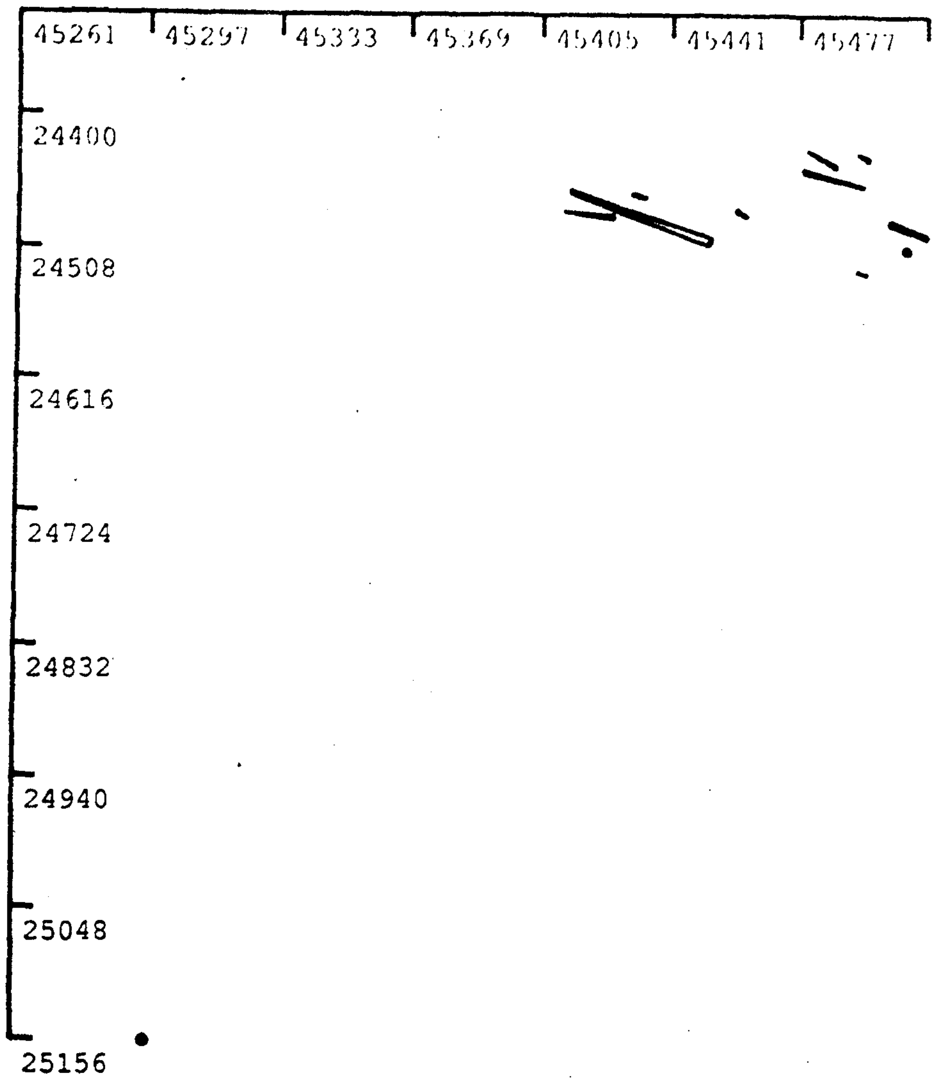

Figure A-6. Linear and online clusters from data set 2.

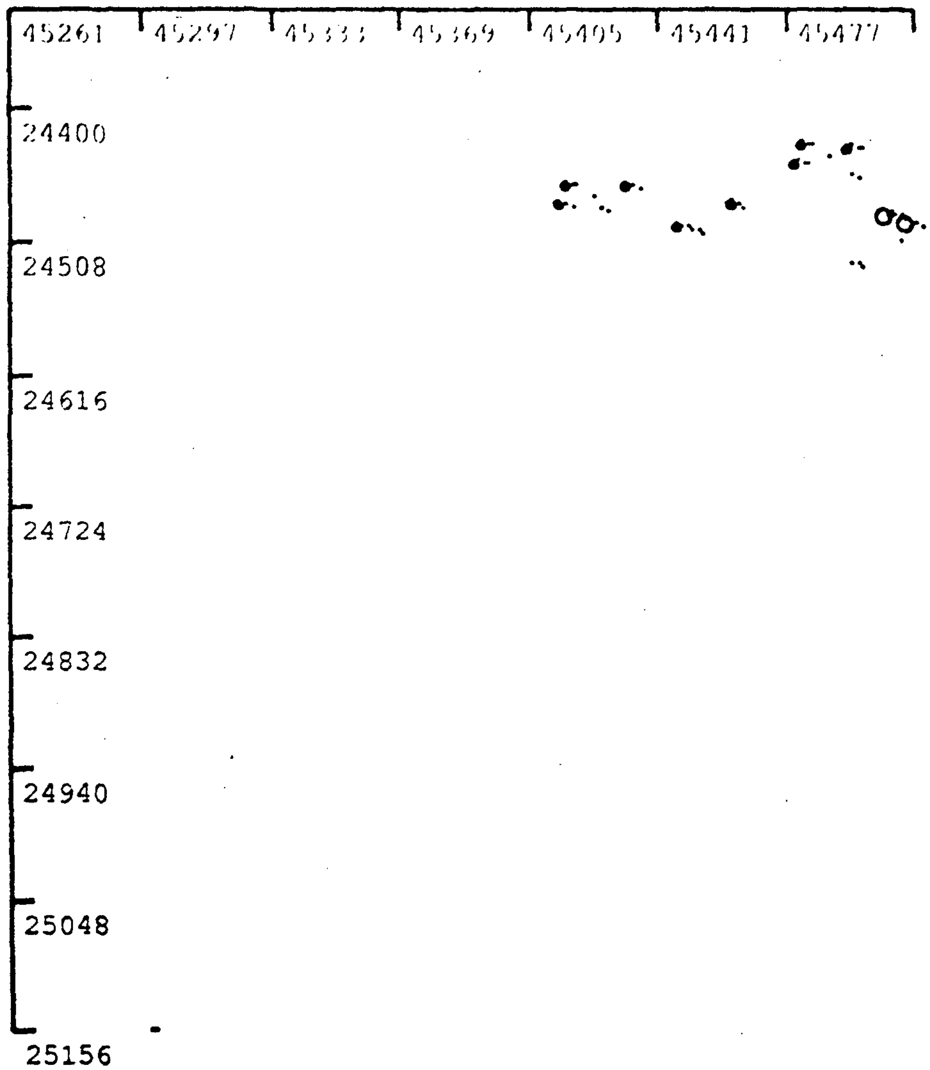

Figure A-7. Circular clustering from data set 2.

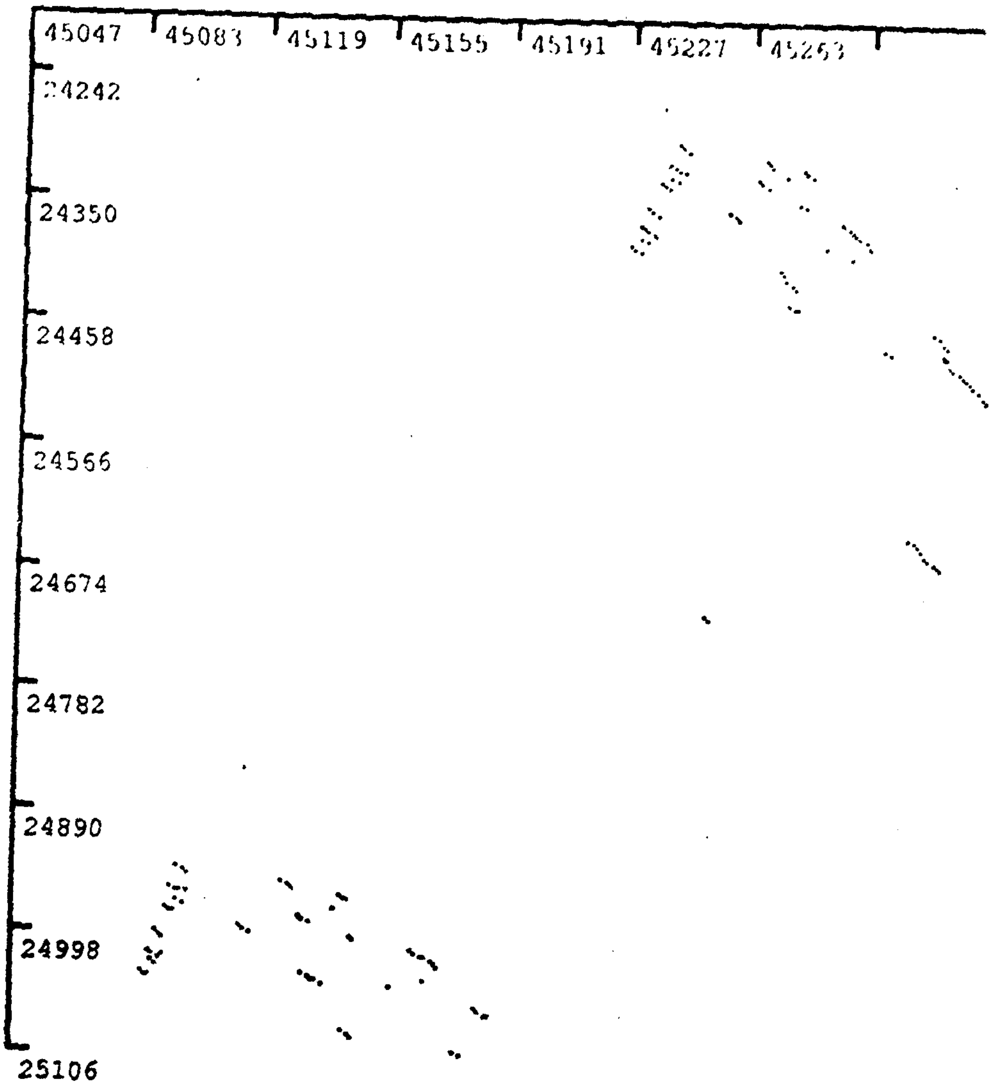

Figure A-8. Input data from data set 3.

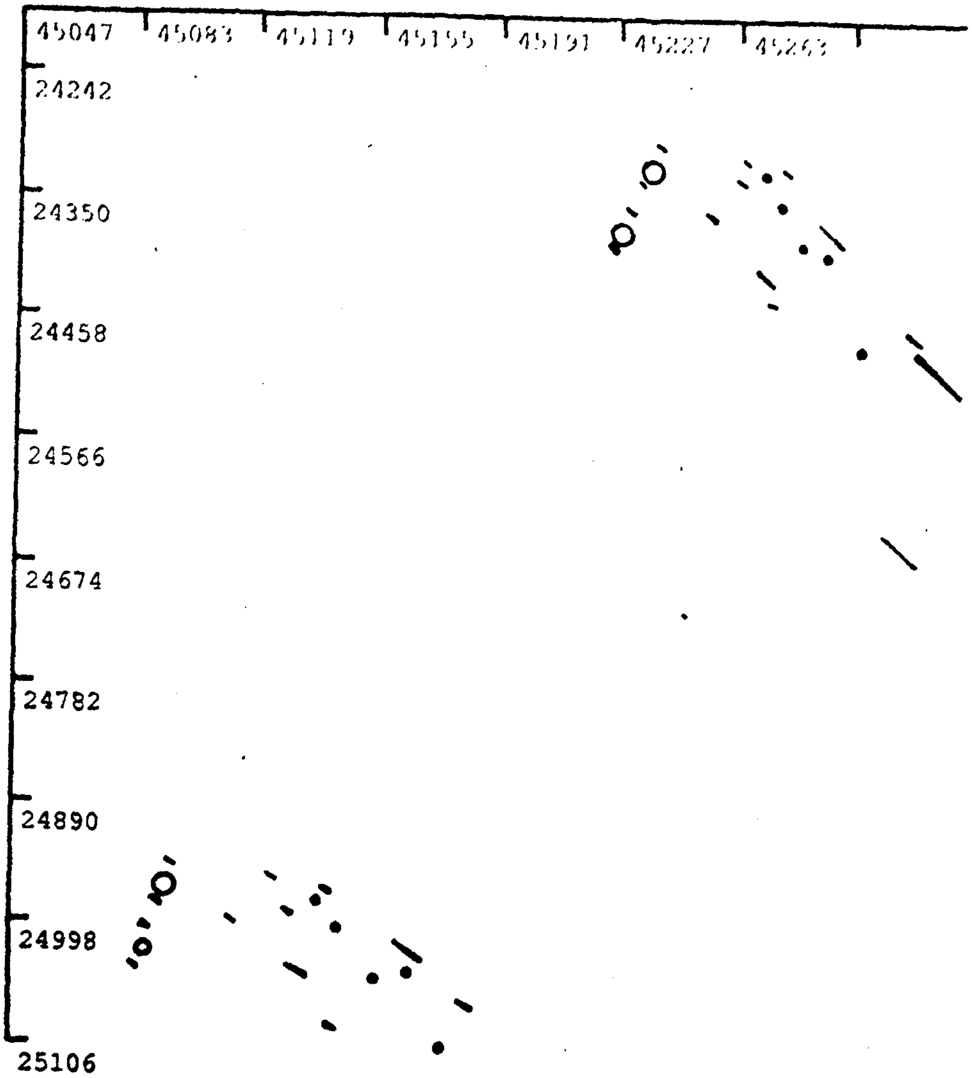

Figure A-9. Initial clusters from data set 3.

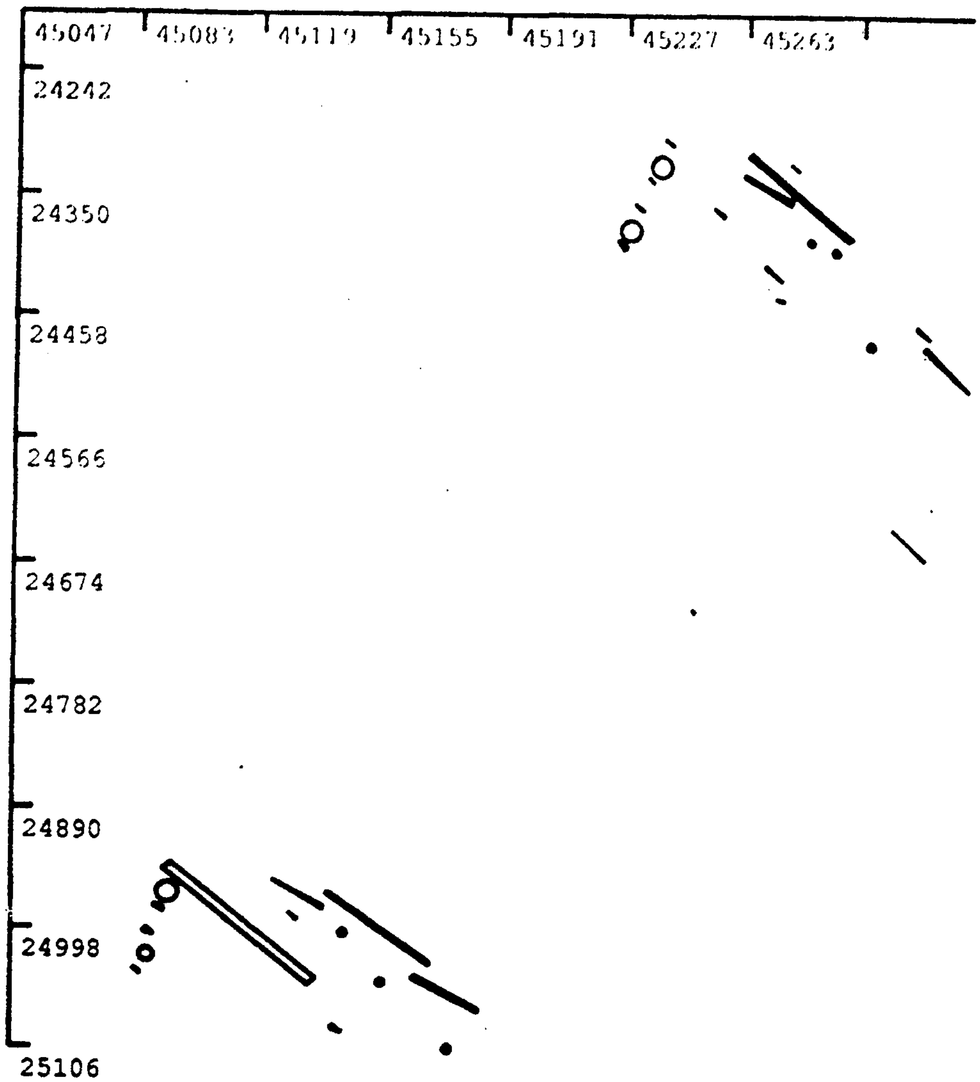

Figure A-10. Linear and online clusters from data set 3.

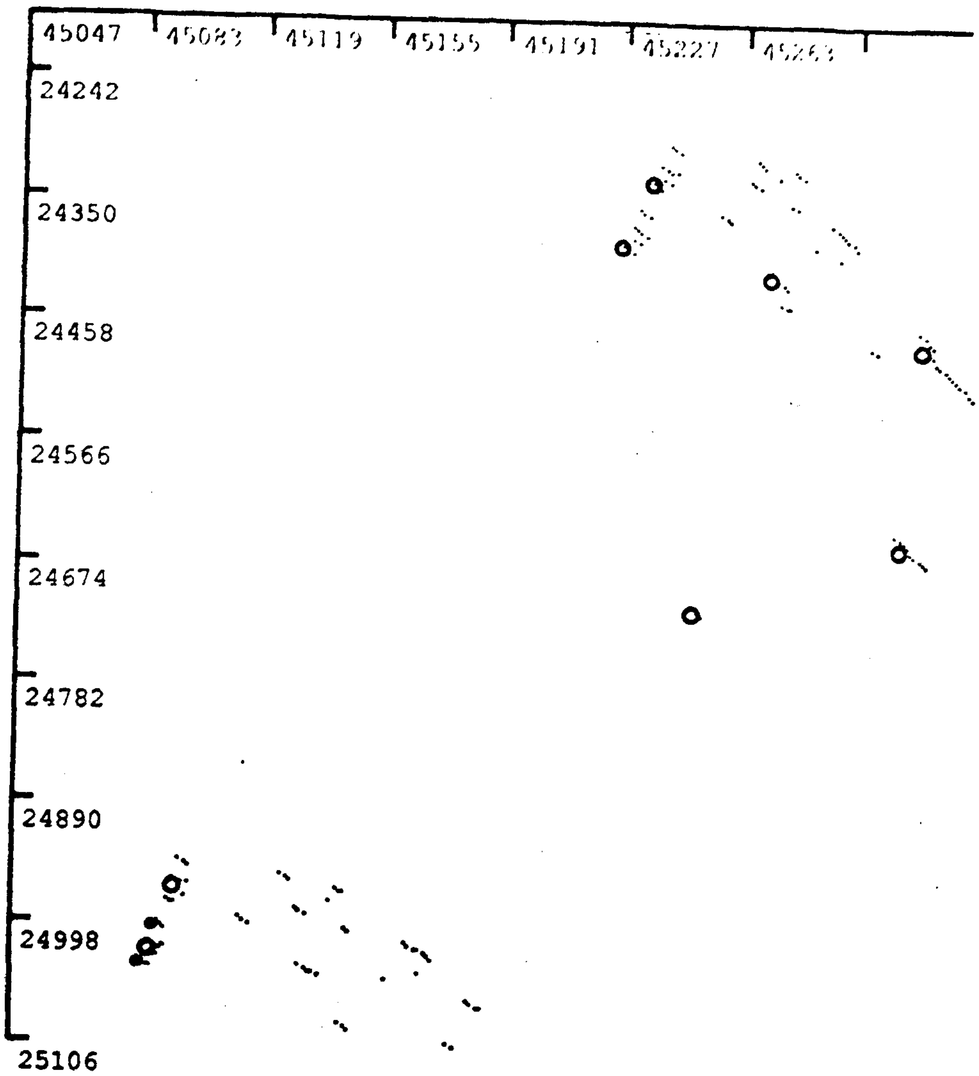

Figure A-11. Circular clustering from data set 3.

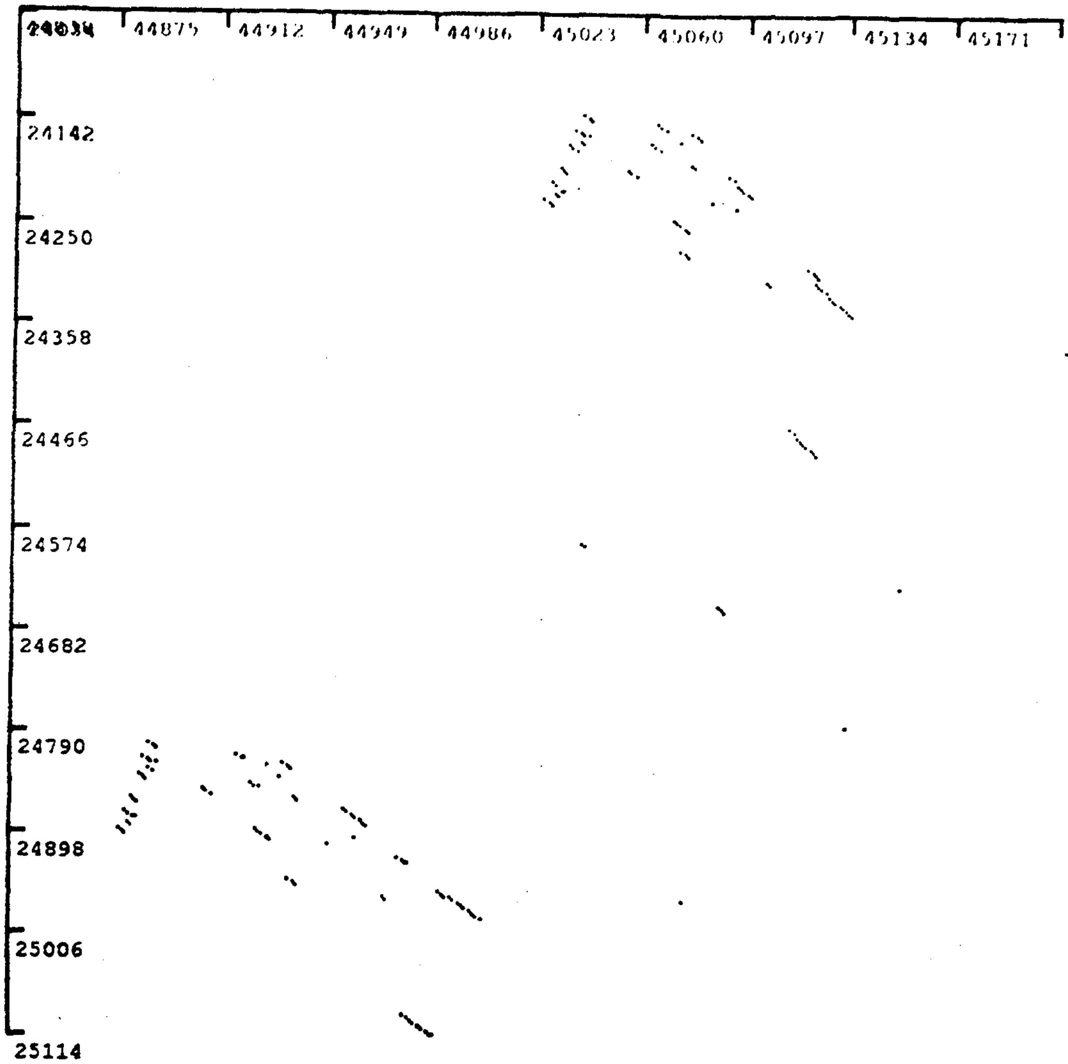

Figure A-12. Input data from data set 4.

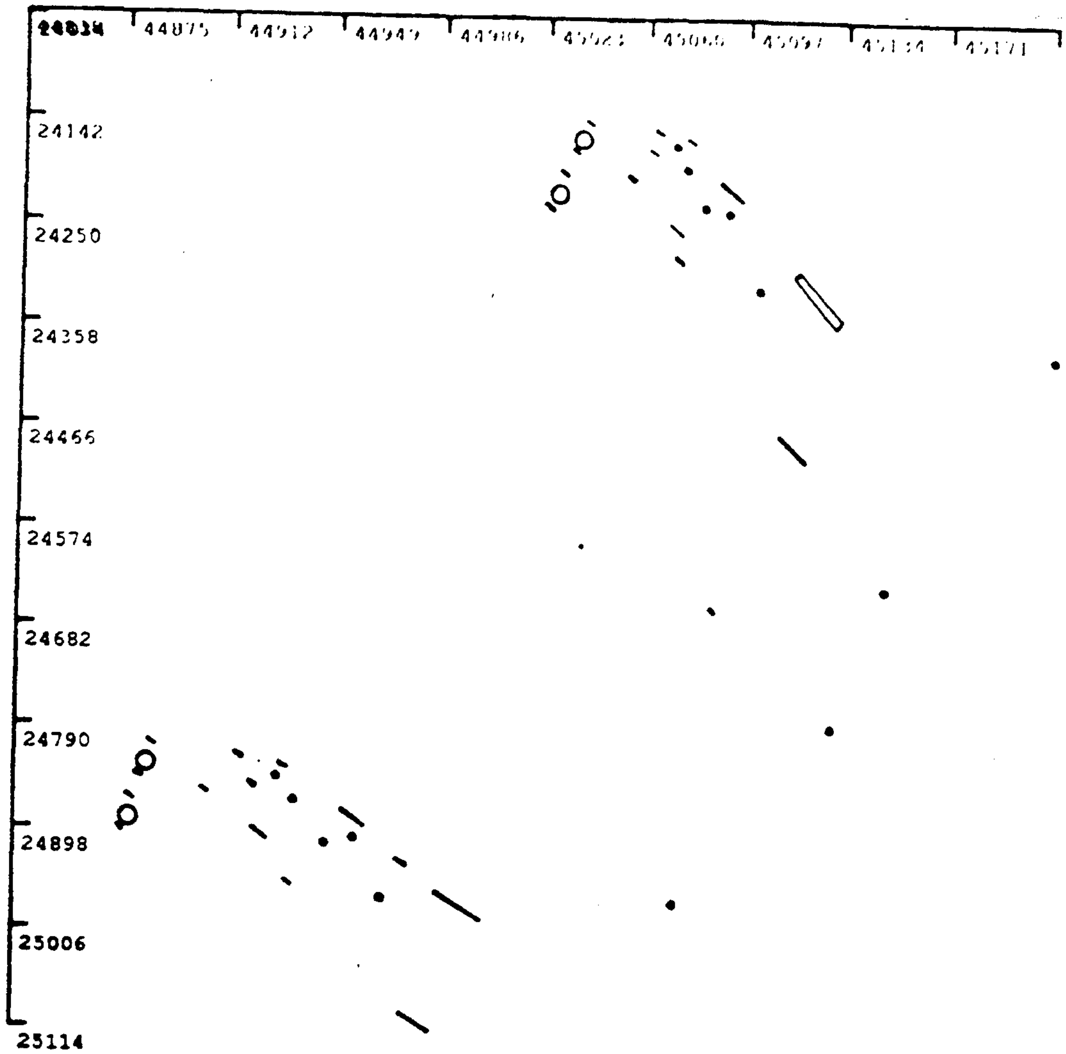

Figure A-13. Initial clusters from data set 4.

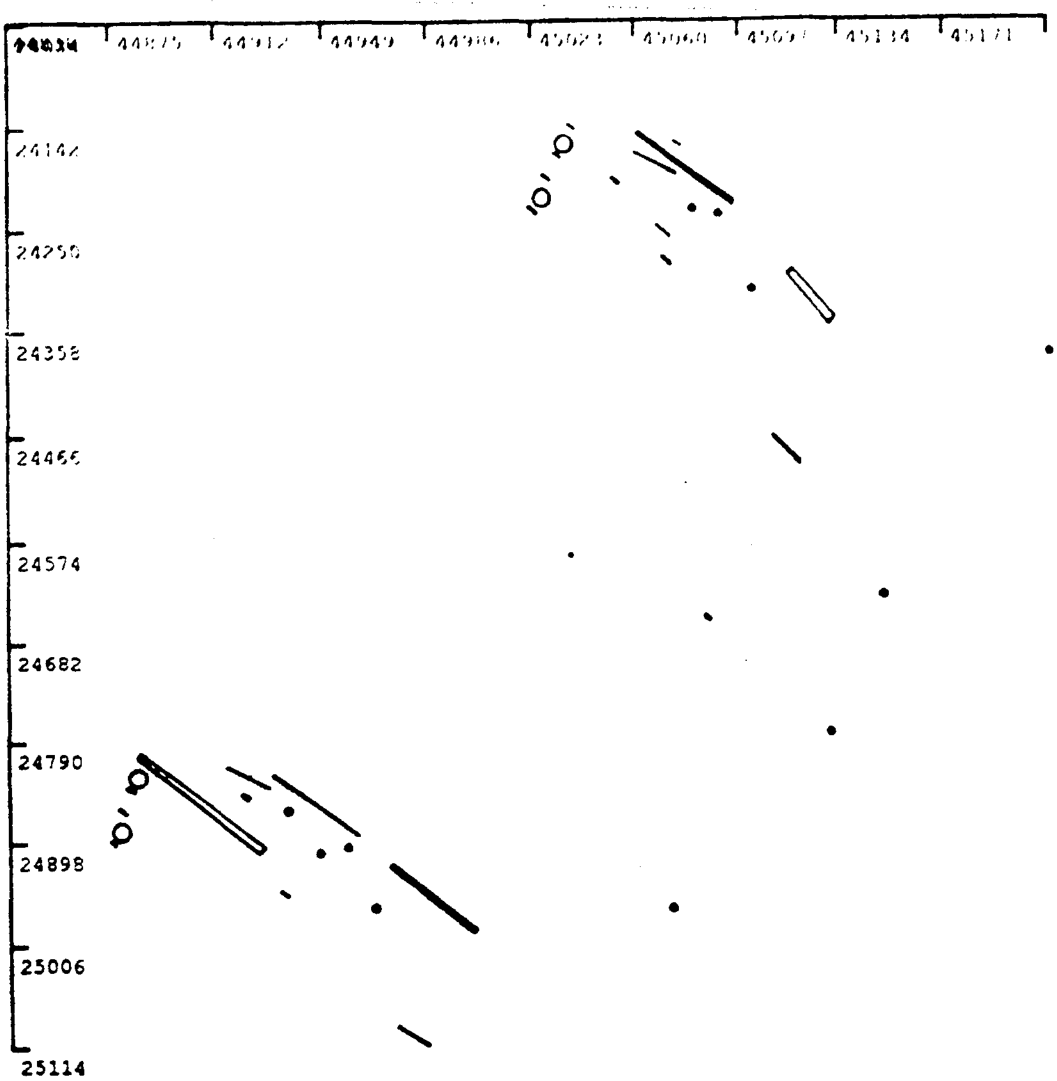

Figure A-14. Linear and online clusters from data set 4.

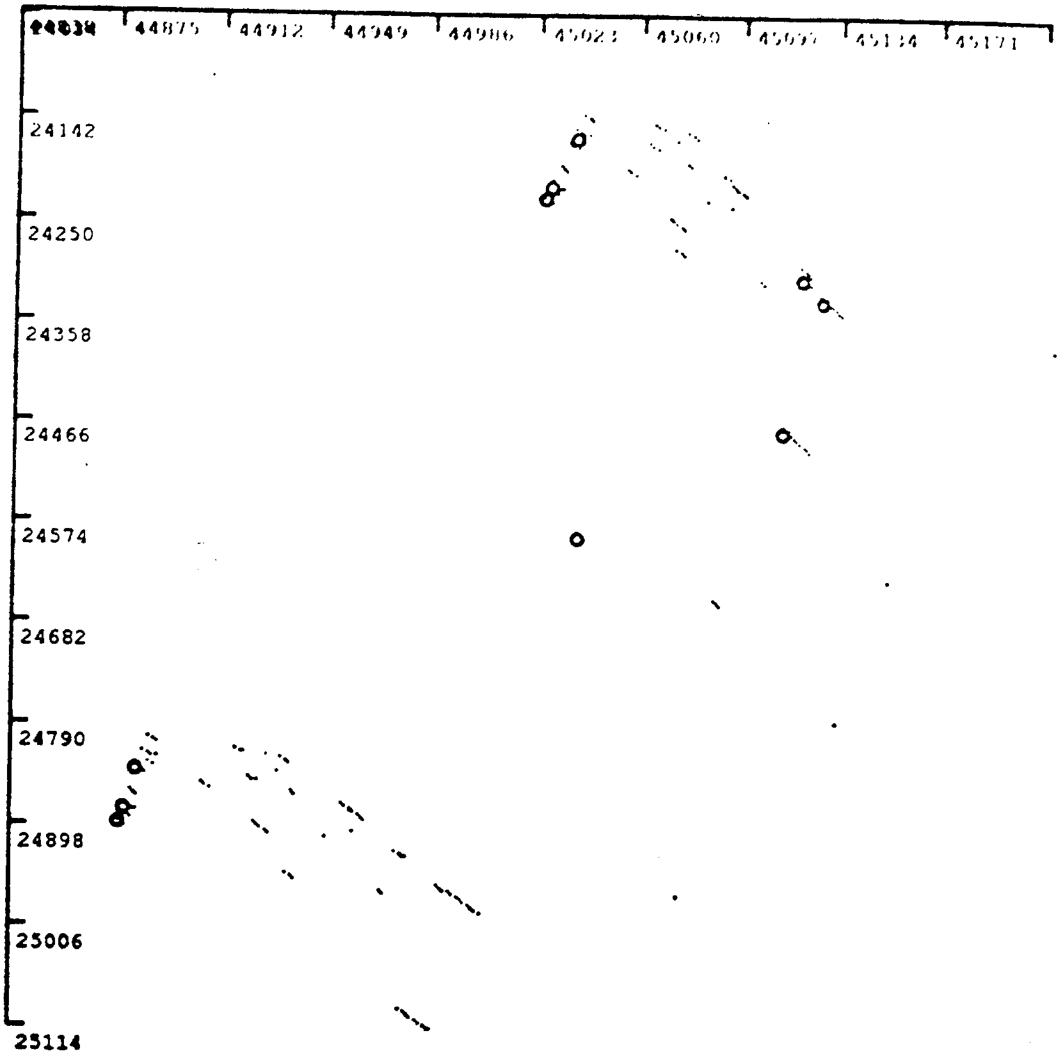

Figure A-15. Circular clustering from data set 4.

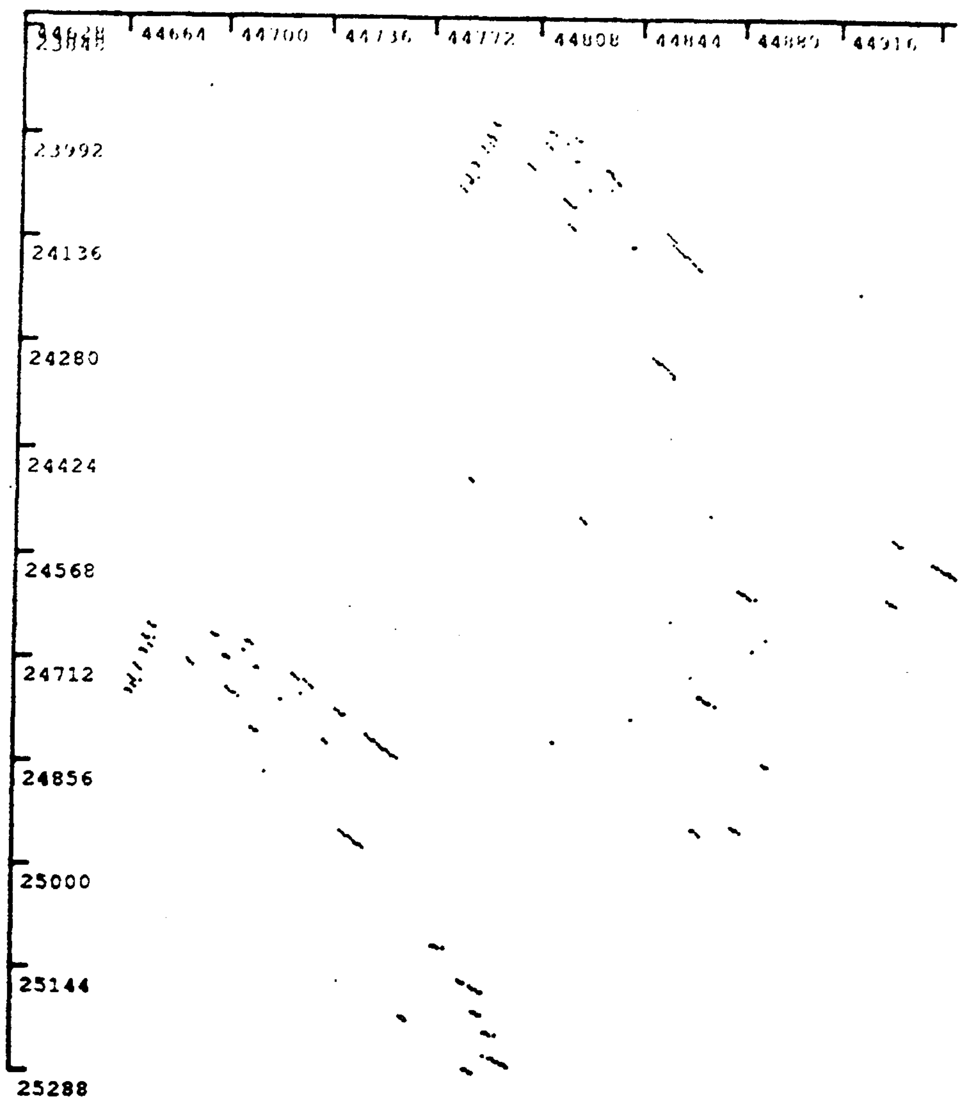

Figure A-16. Input data from data set 5.

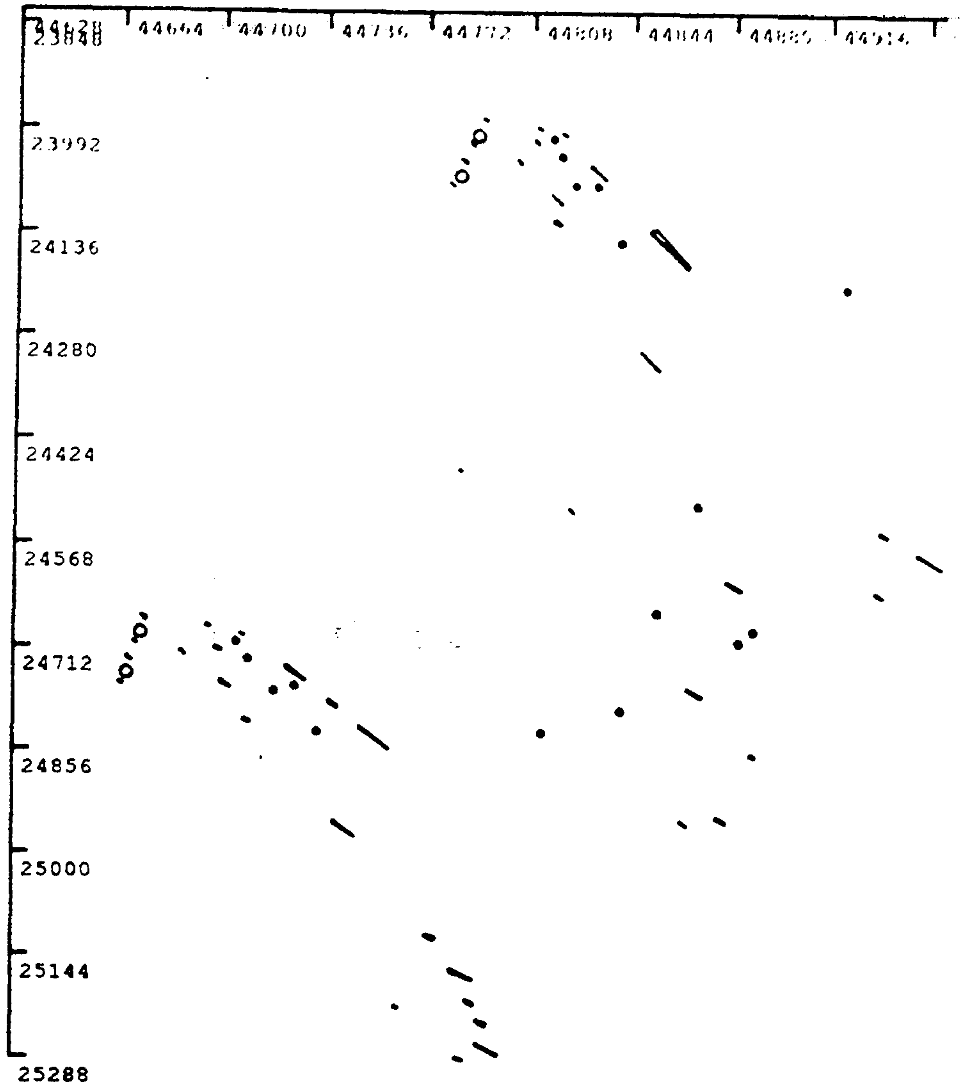

Figure A-17. Initial clusters from data set 5.

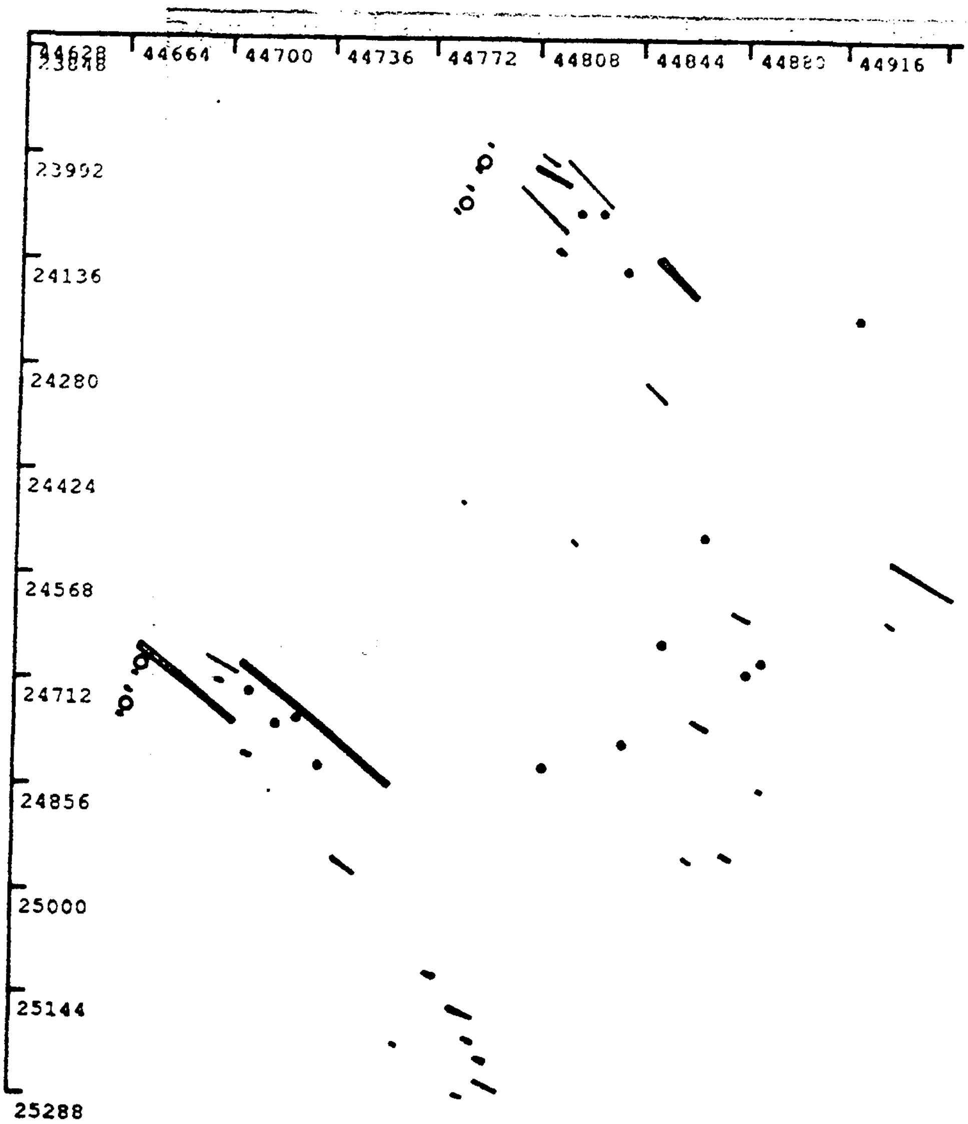

Figure A-18. Linear and online clusters from data set 5.

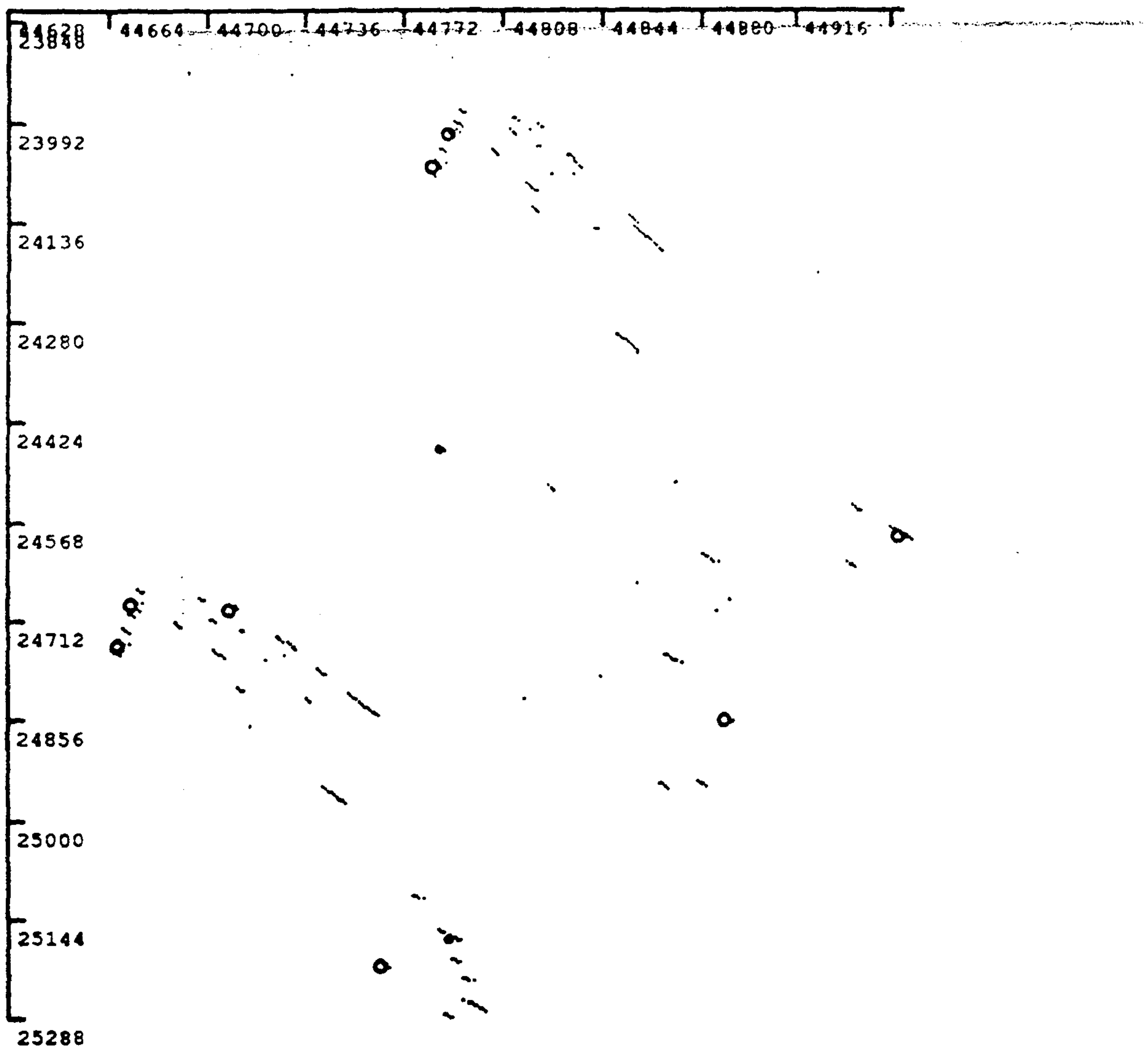

Figure A-19. Circular clustering from data set 5.

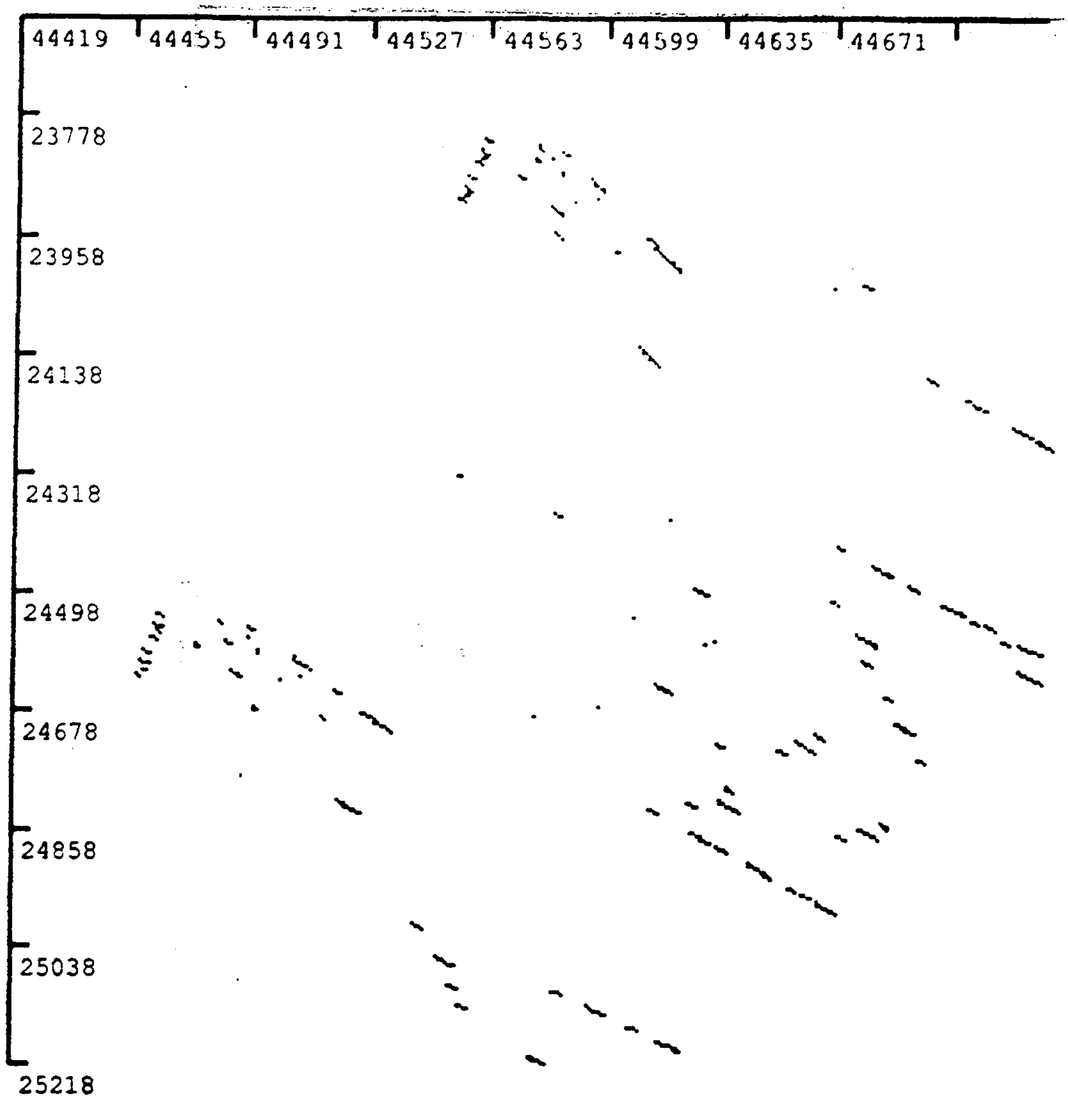

Figure A-20. Input data from data set 6.

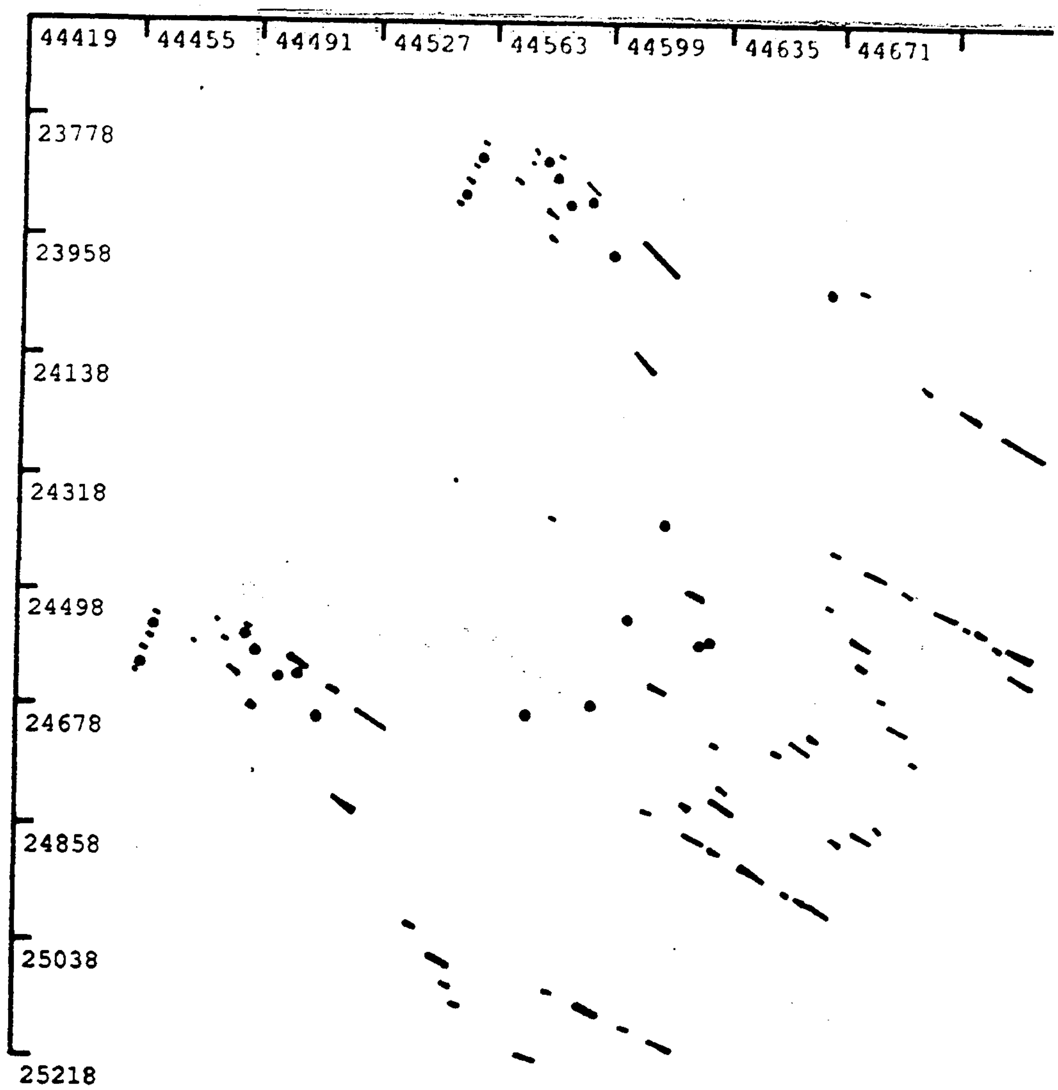

Figure A-21. Initial clusters from data set 6.

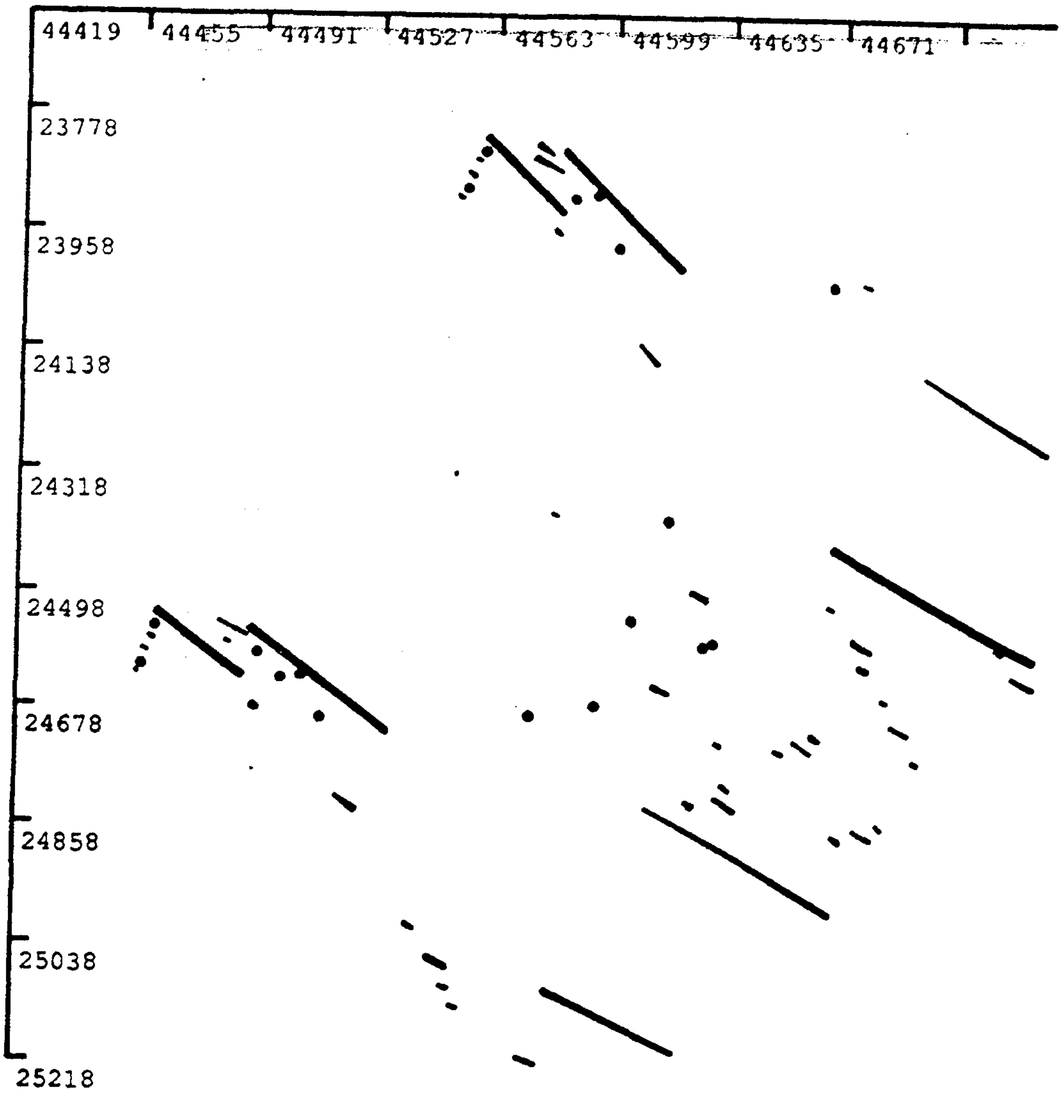

Figure A-22. Linear and online clusters from data set 6.

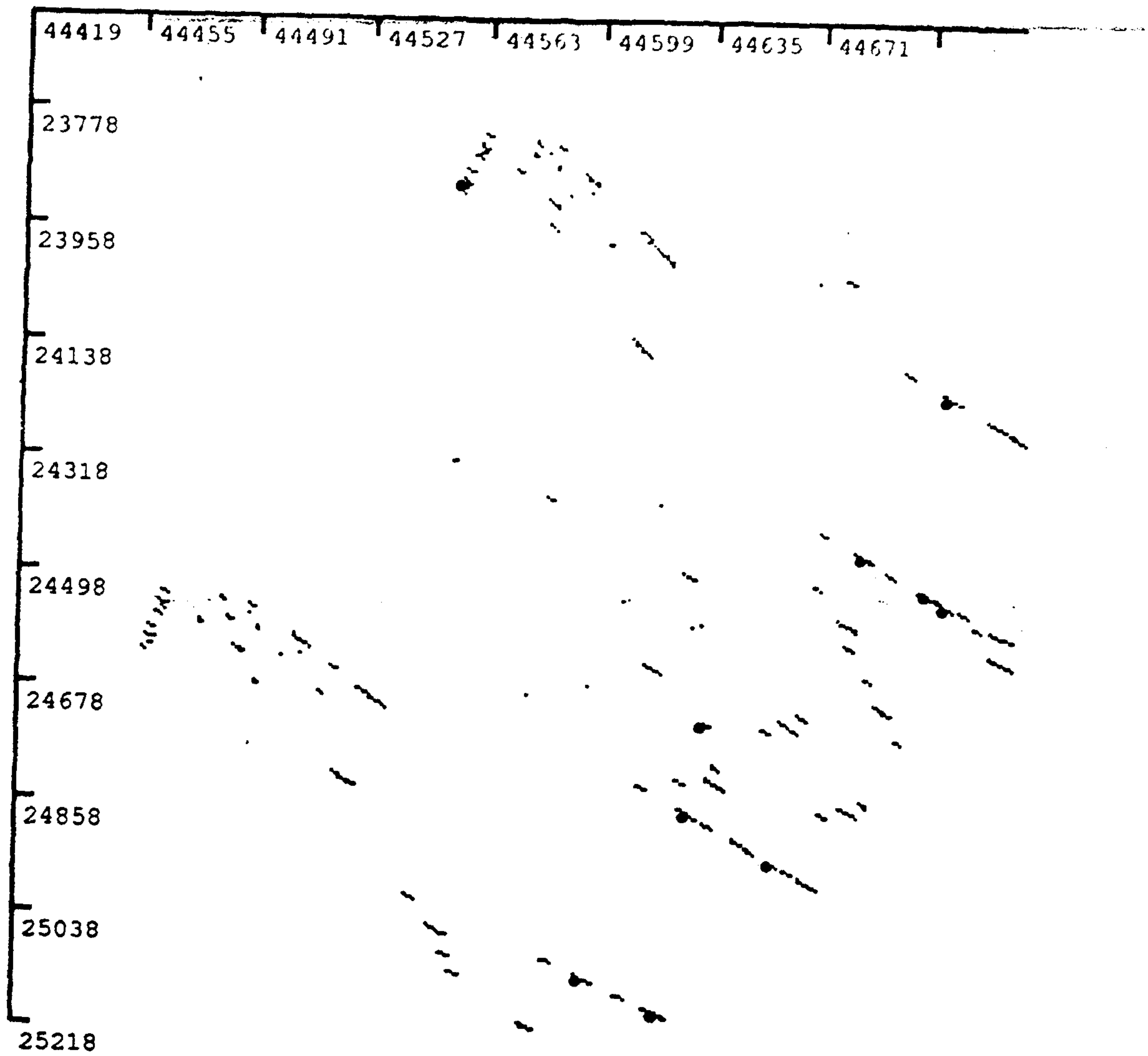

Figure A-23. Circular clustering from data set 6.

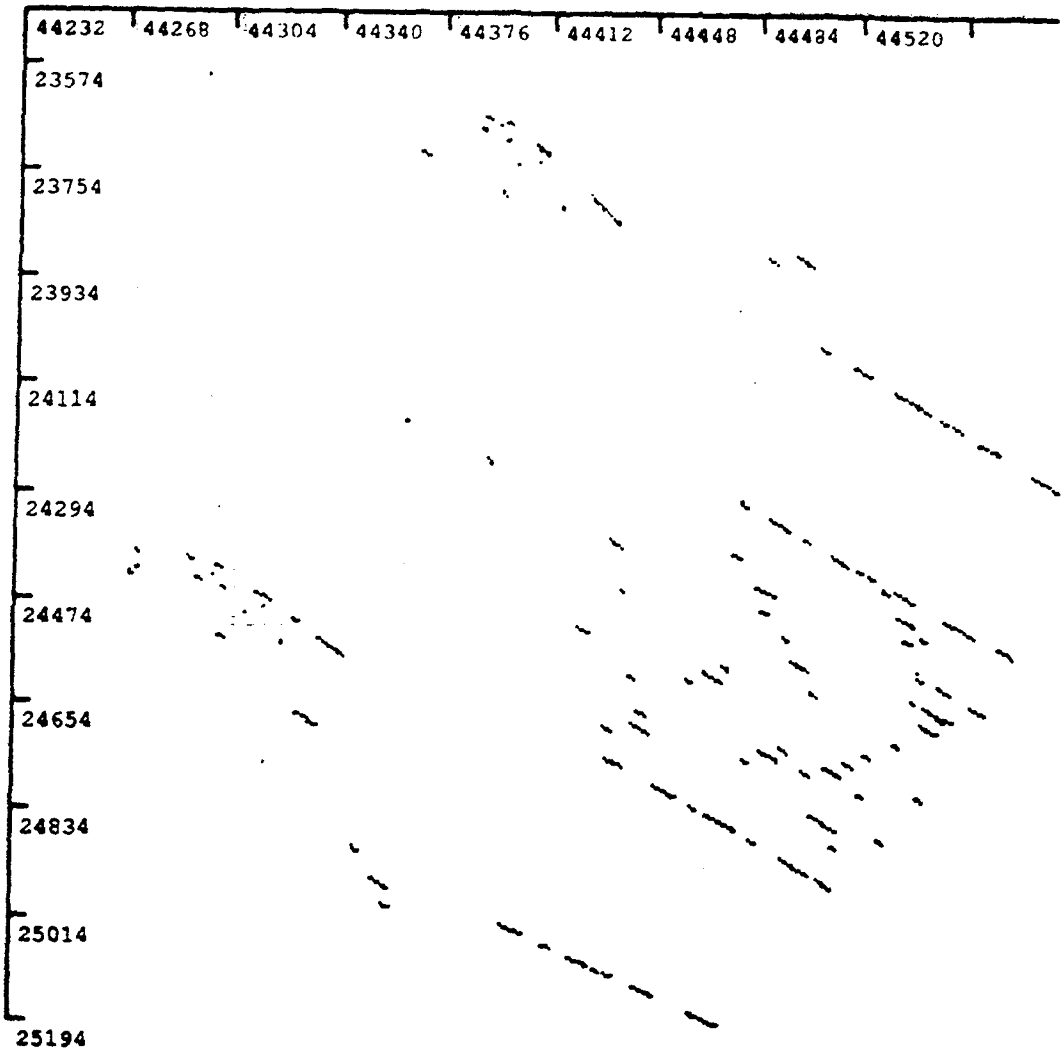

Figure A-24. Input data from data set 7.

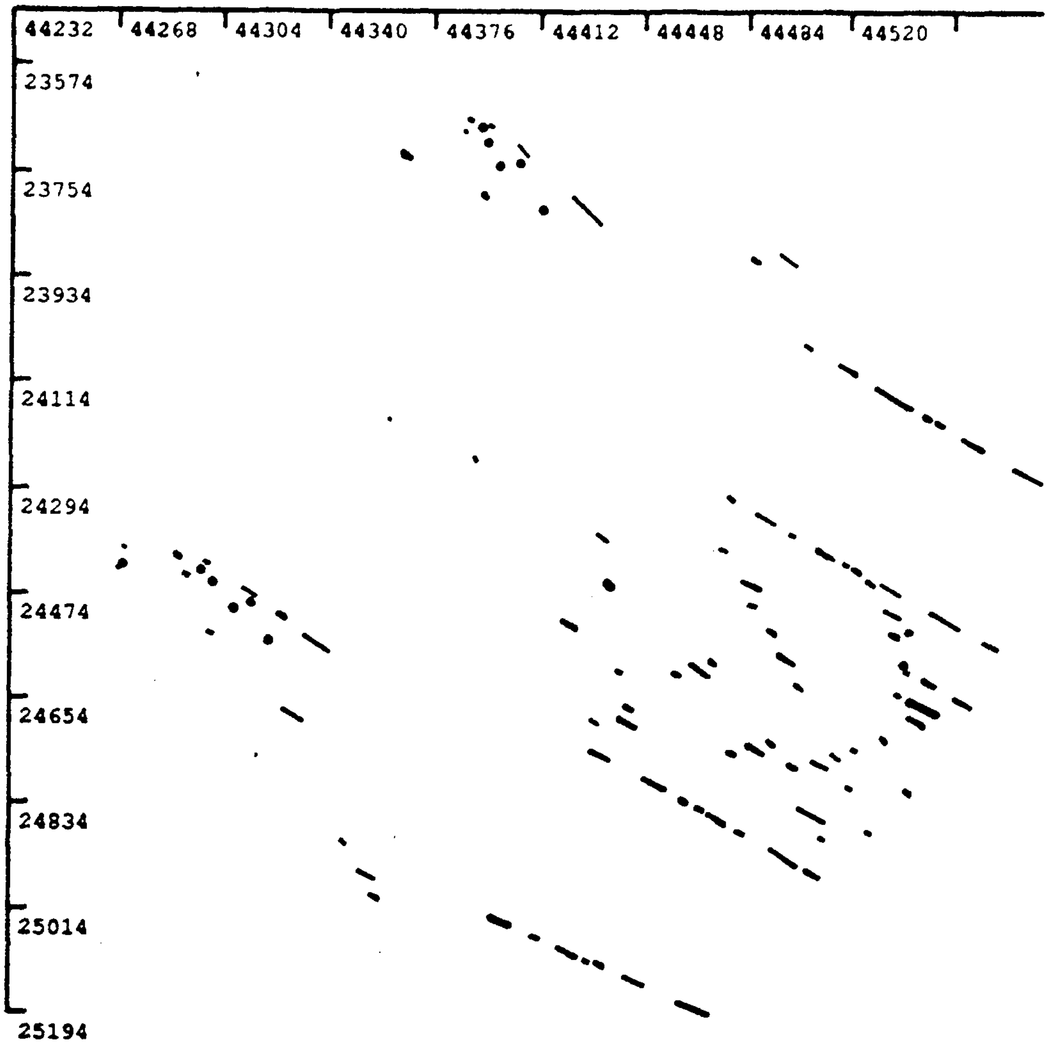

Figure A-25. Initial clusters from data set 7.

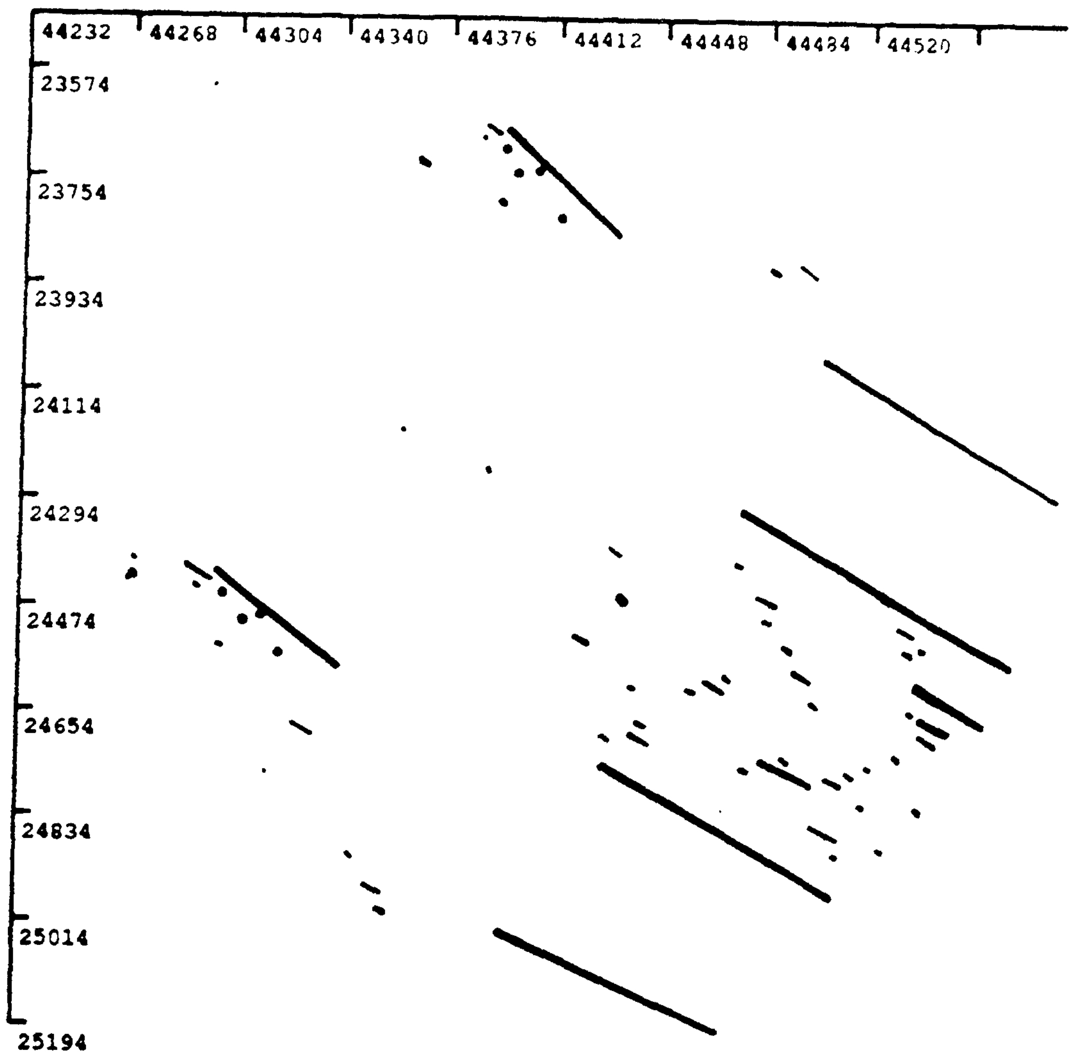

Figure A-26. Linear and online clusters from data set 7.

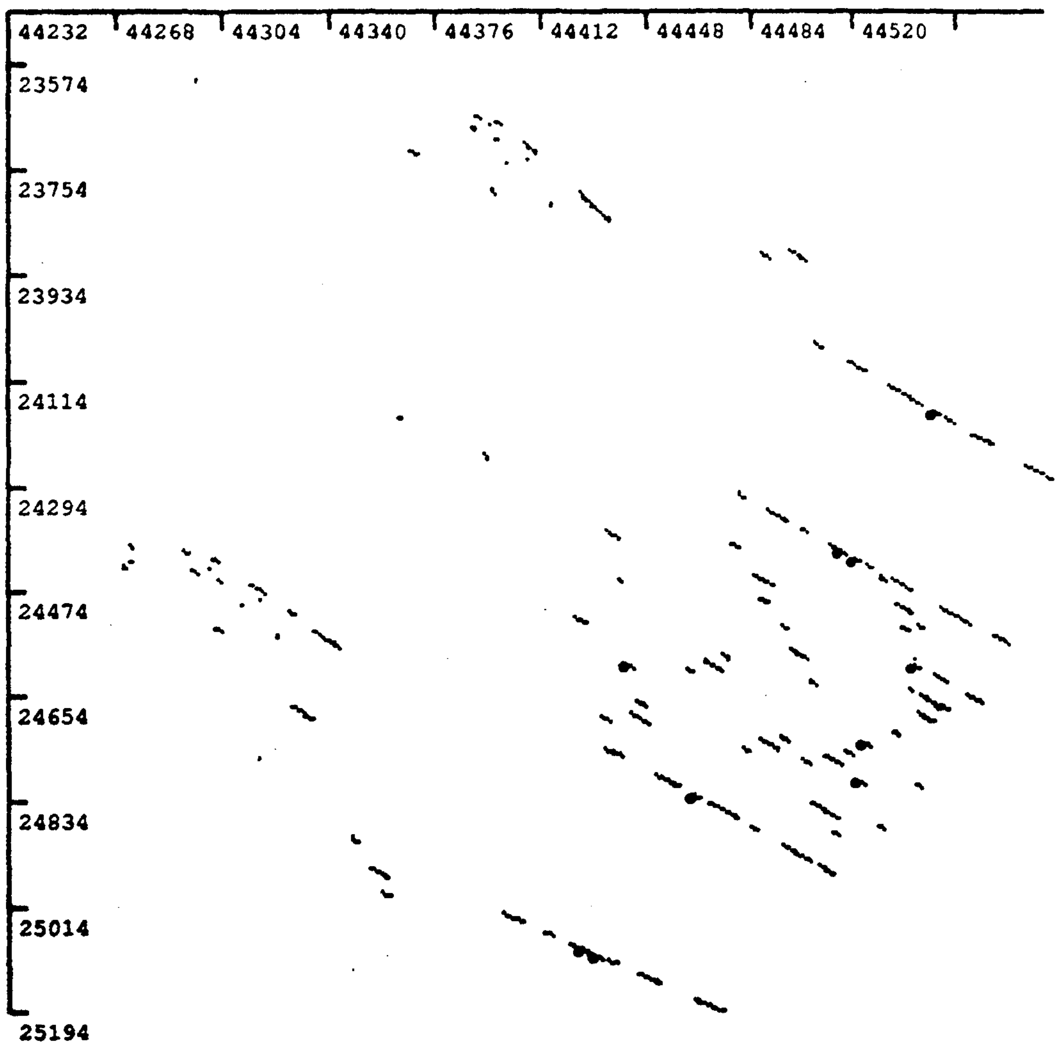

Figure A-27. Circular clustering from data set 7.

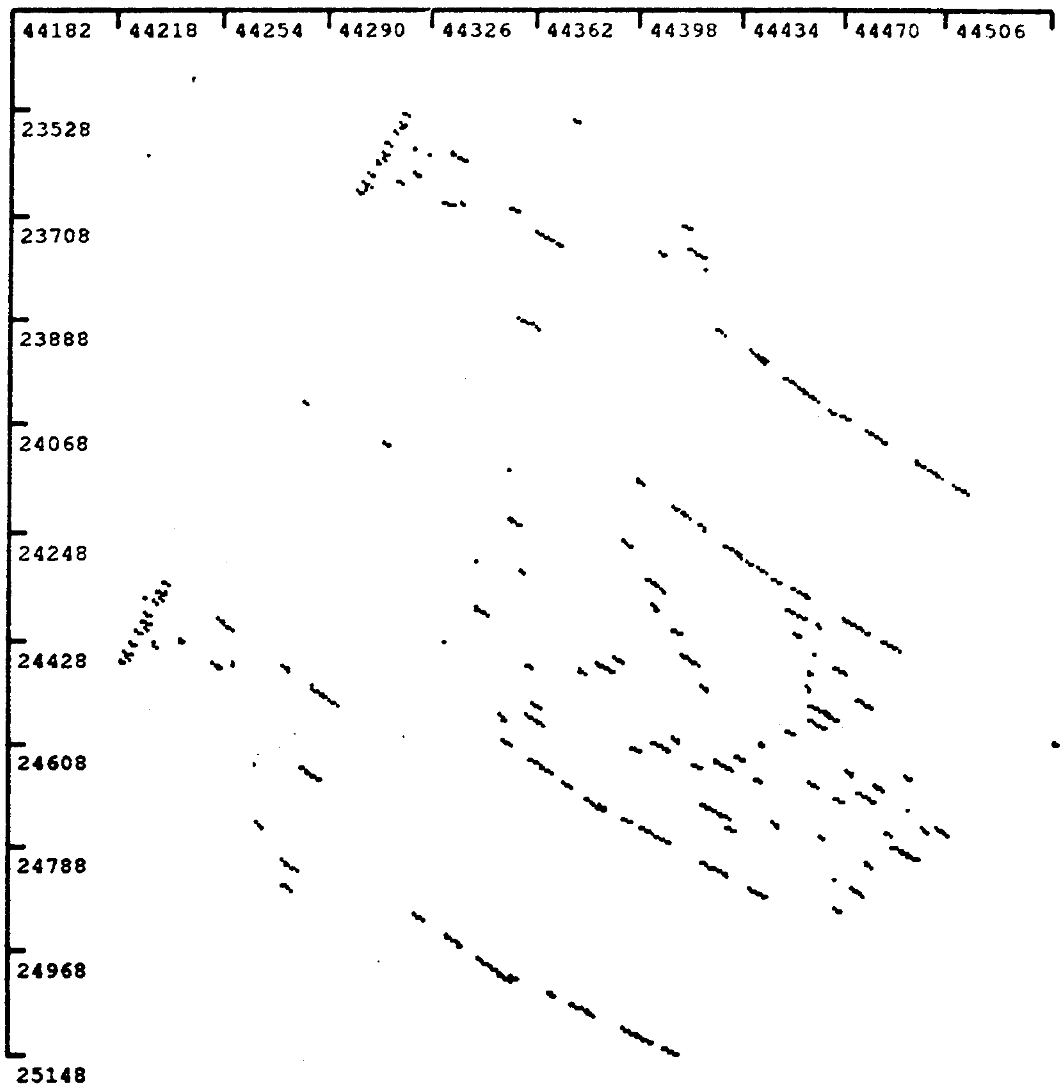

Figure A-28. Input data from data set 8.

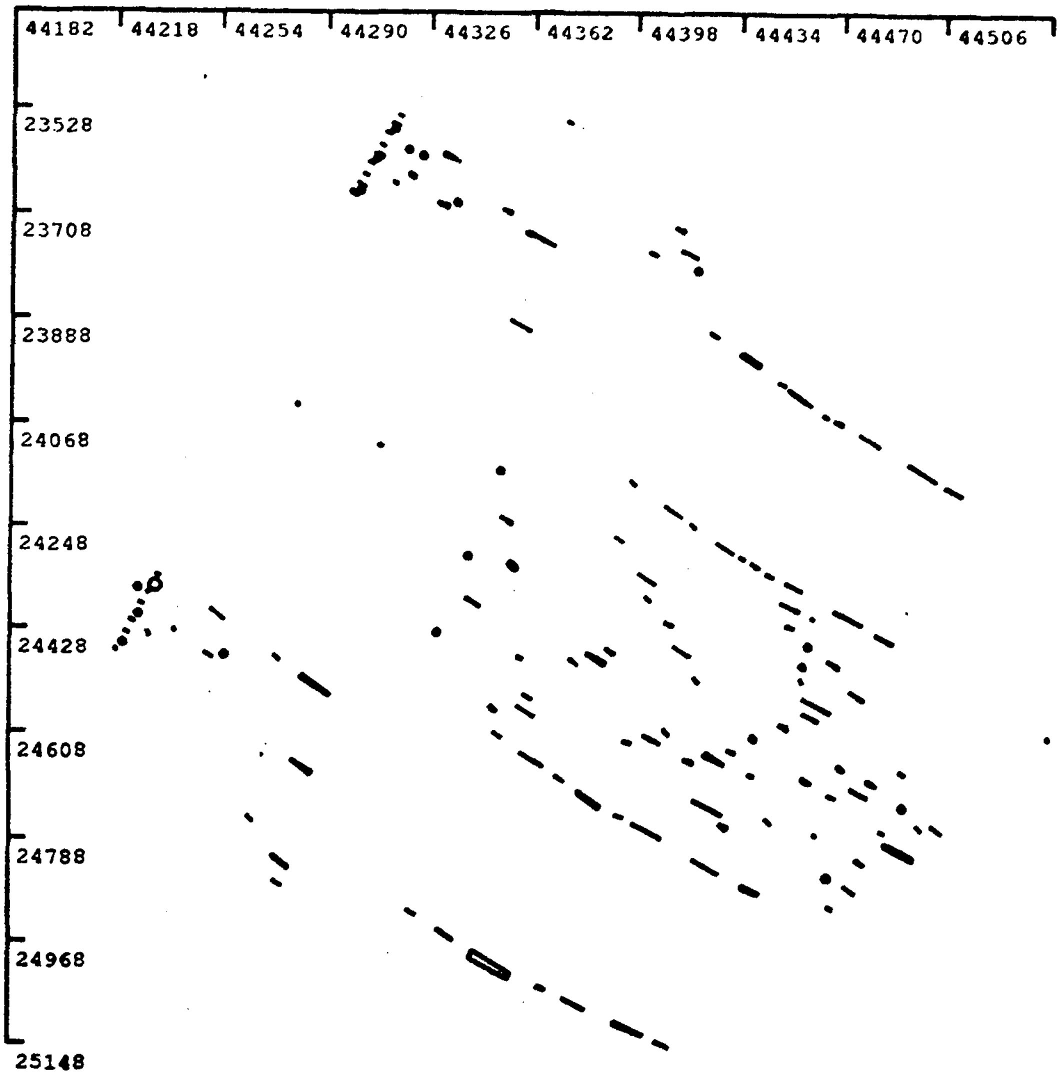

Figure A-29. Initial clusters from data set 8.

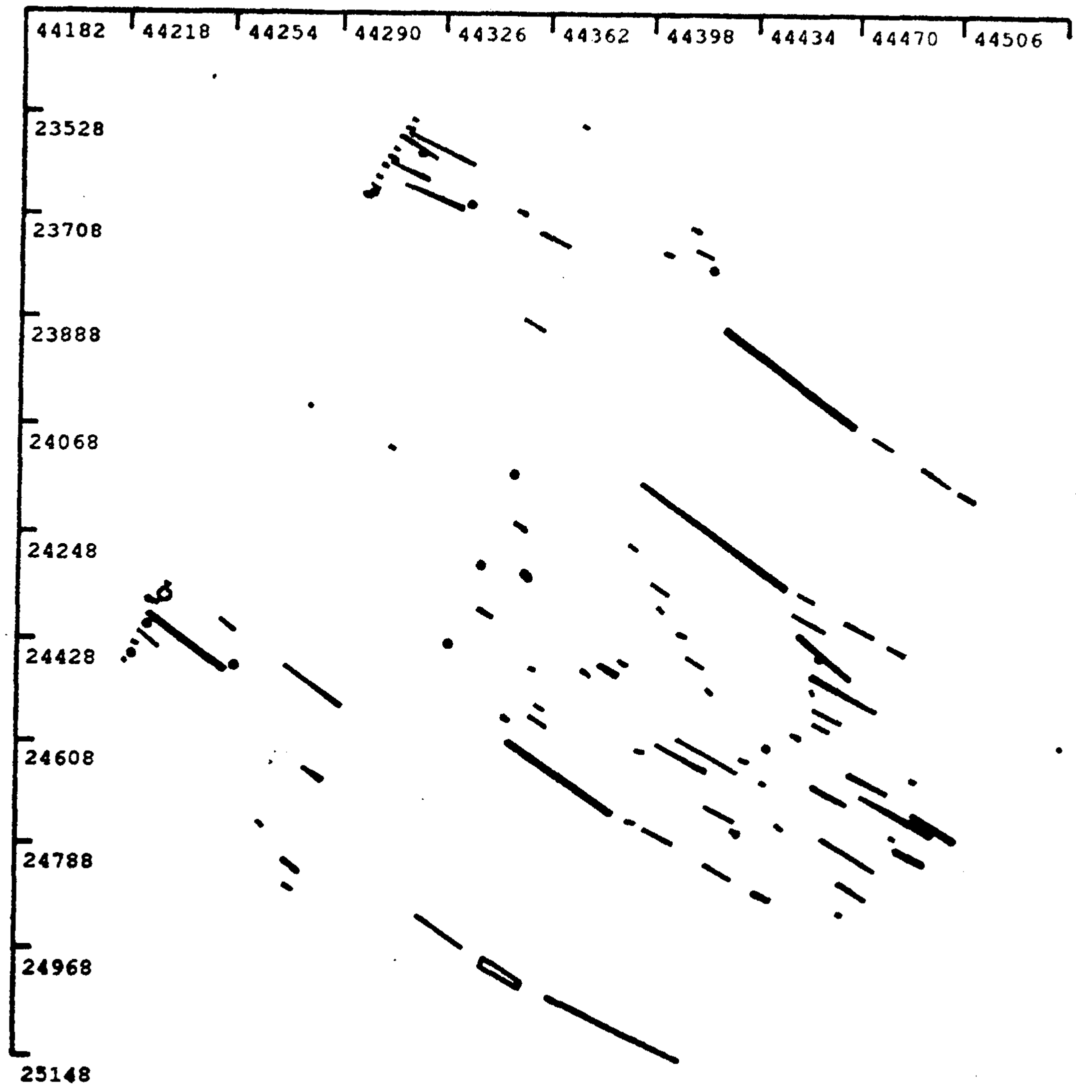

Figure A-30. Linear and online clusters from data set 8.

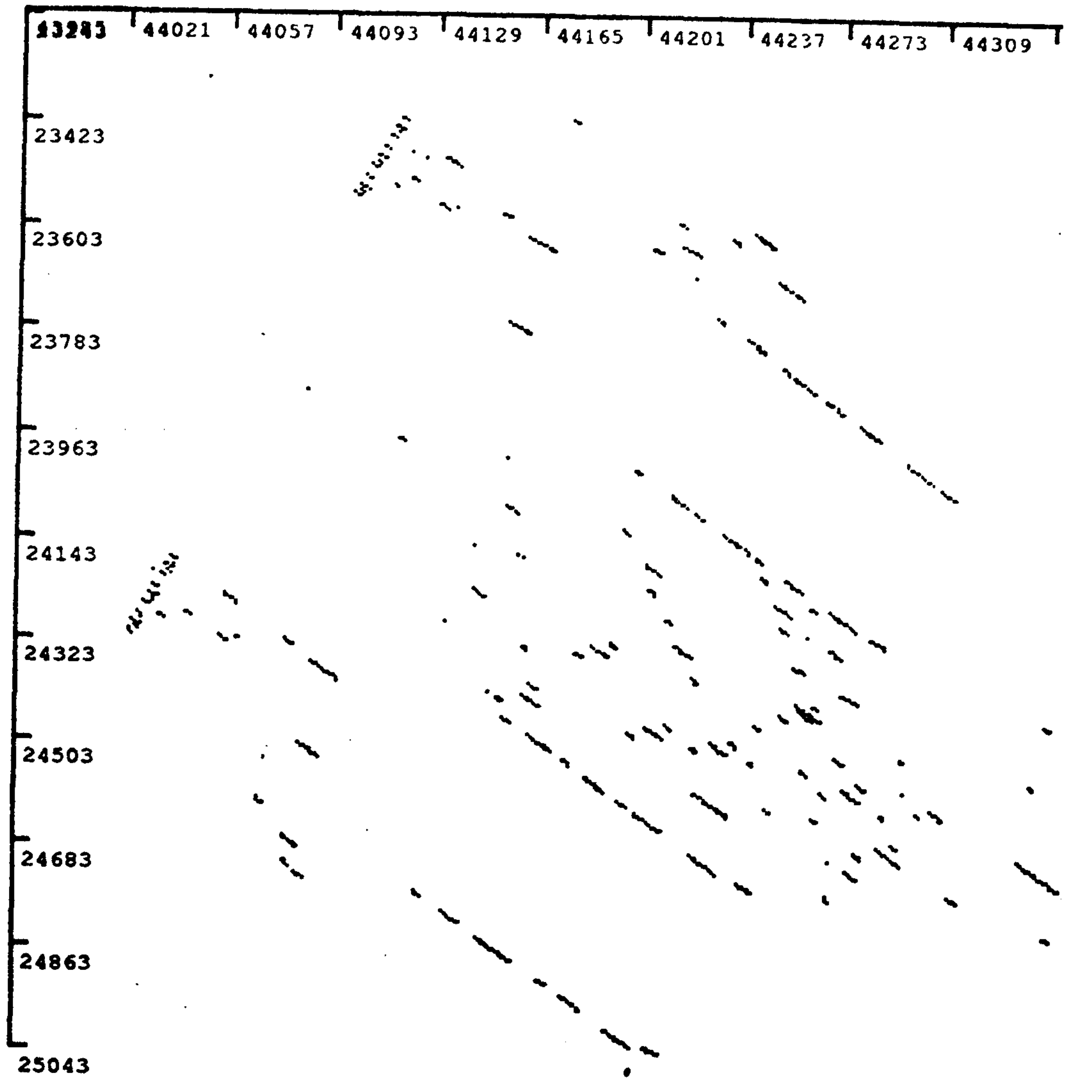

Figure A-31. Input data from data set 9.

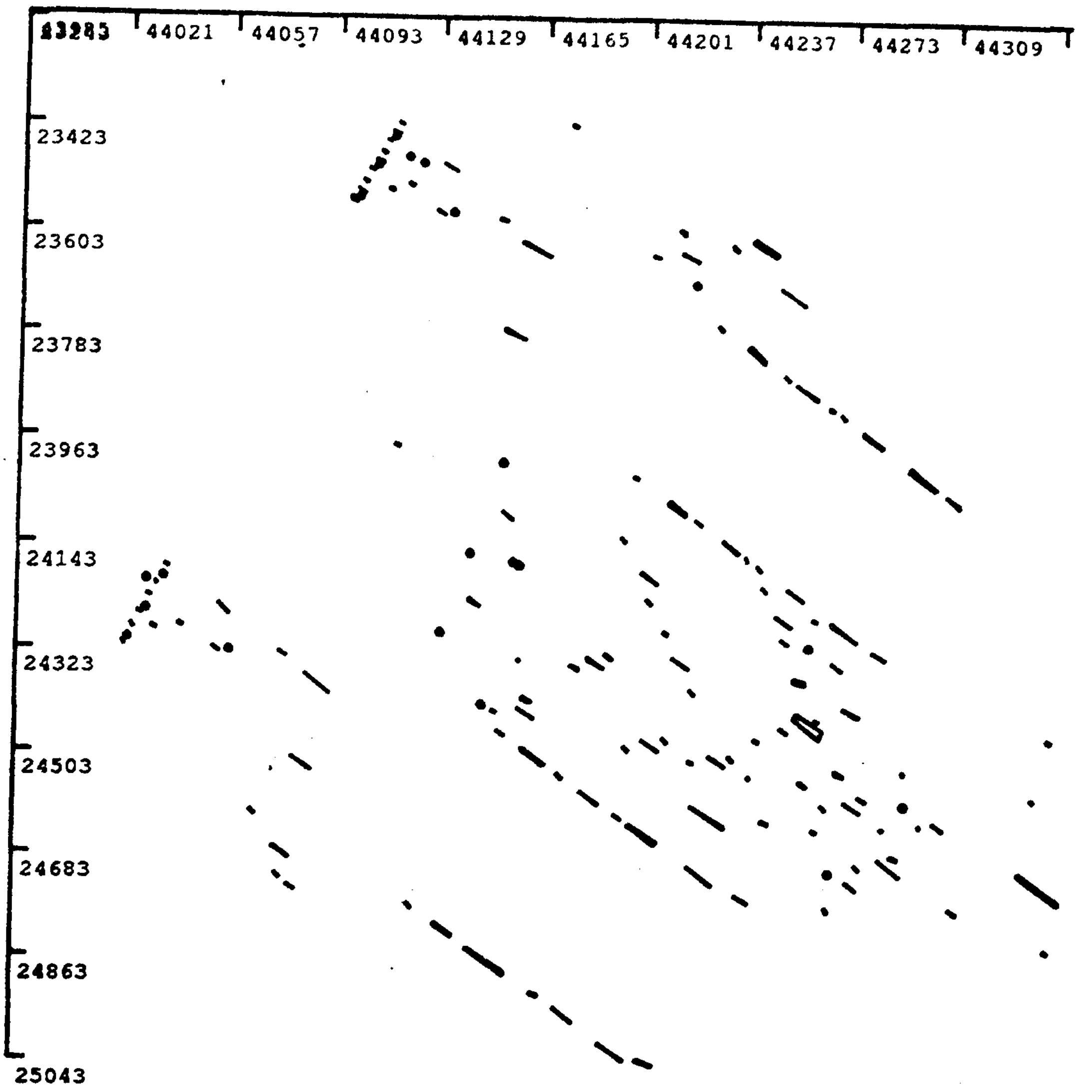

Figure A-32. Initial clusters from data set 9.

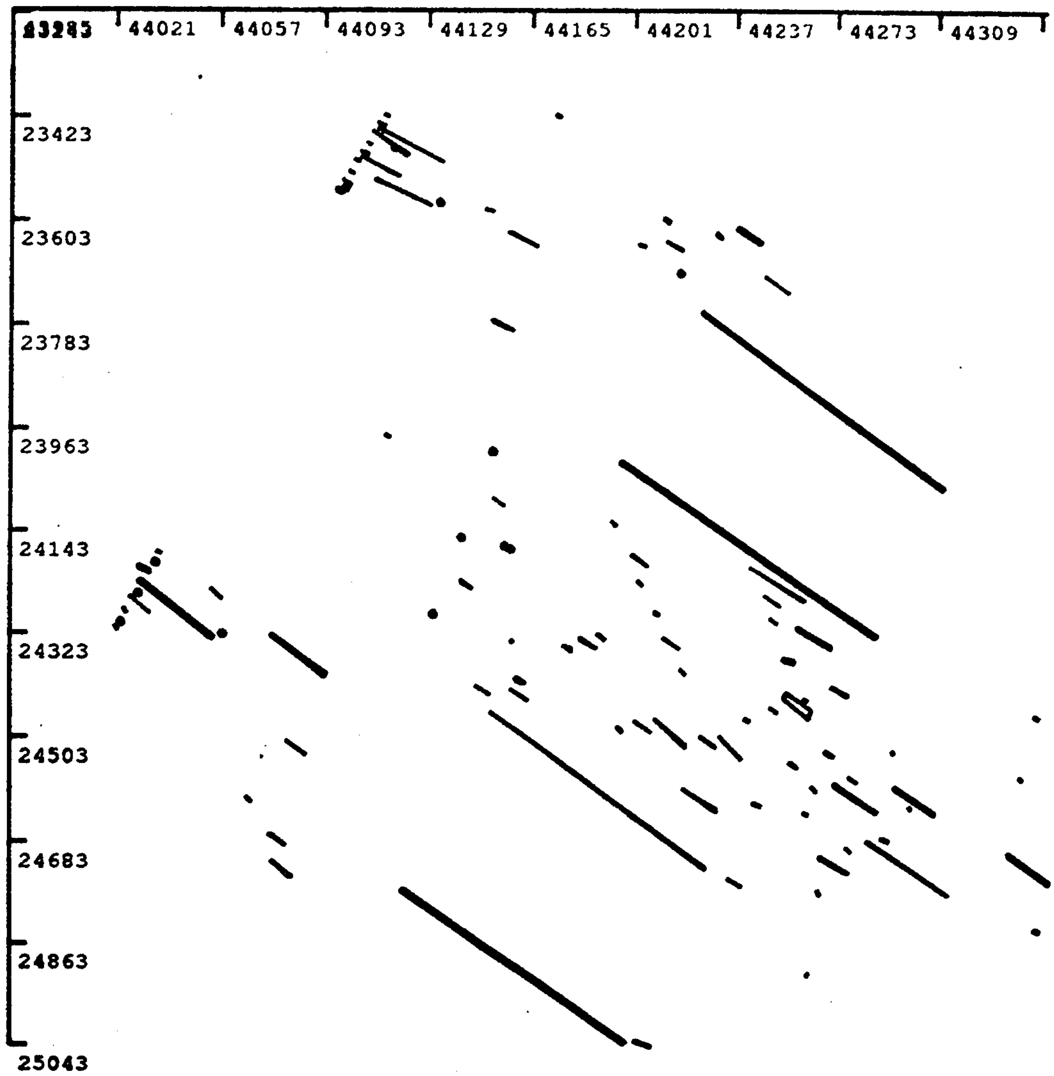

Figure A-33. Linear and online clusters from data set 9.

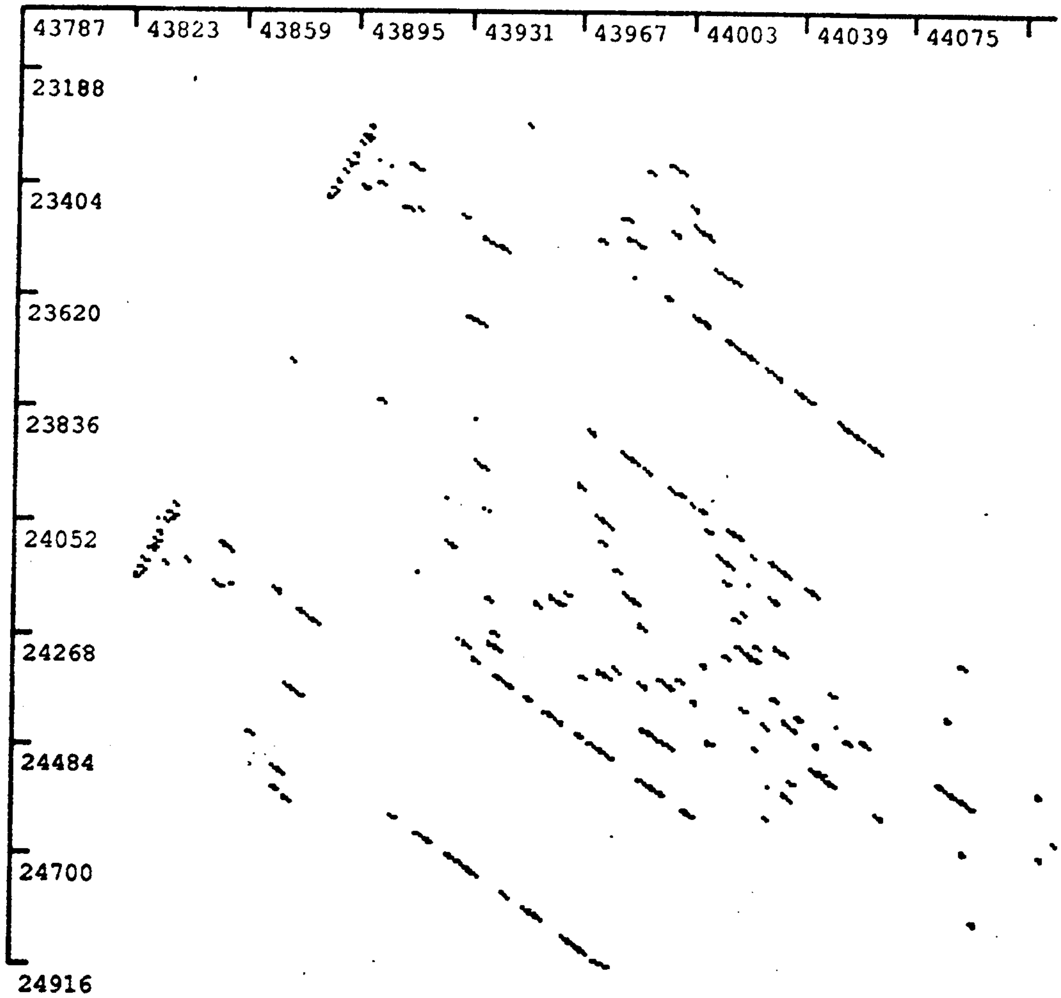

Figure A-34. Input data from data set 10.

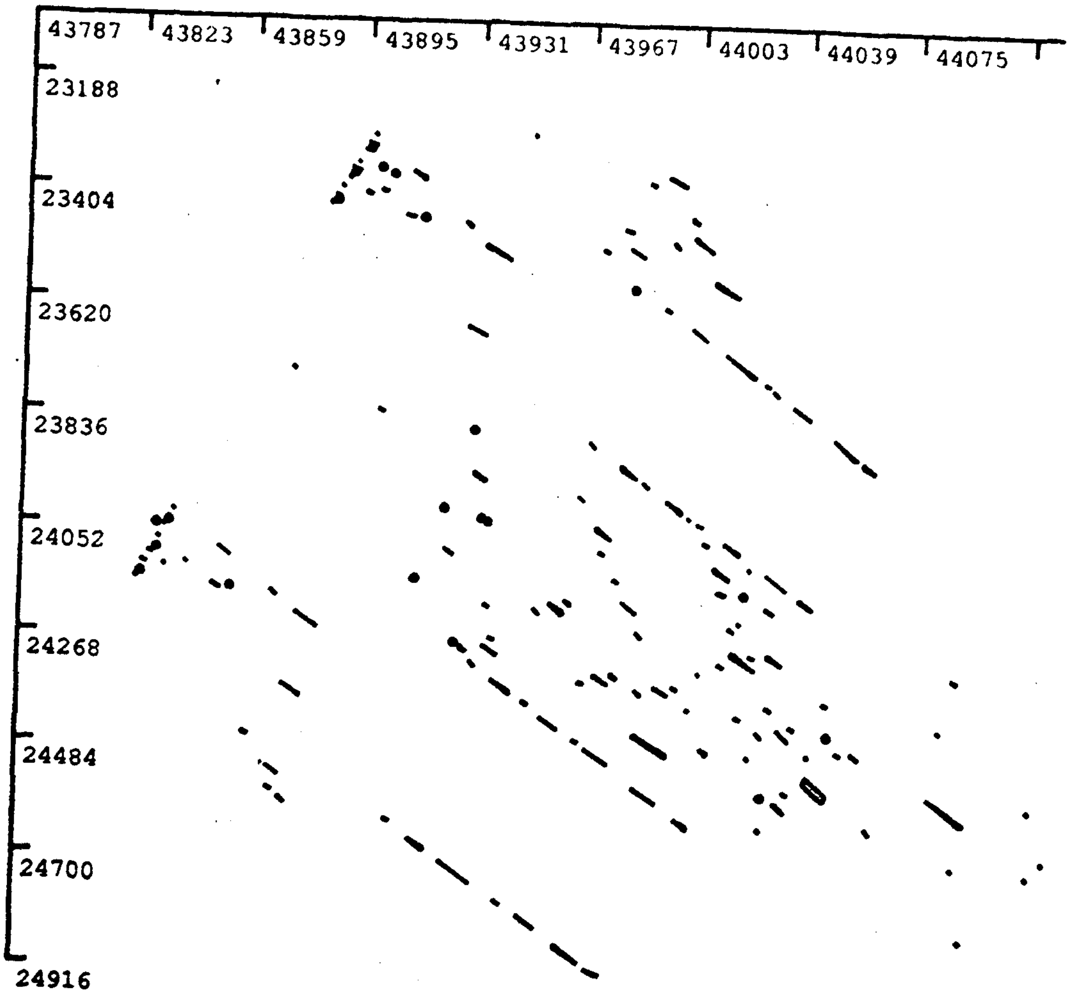

Figure A-35. Initial clusters from data set 10.

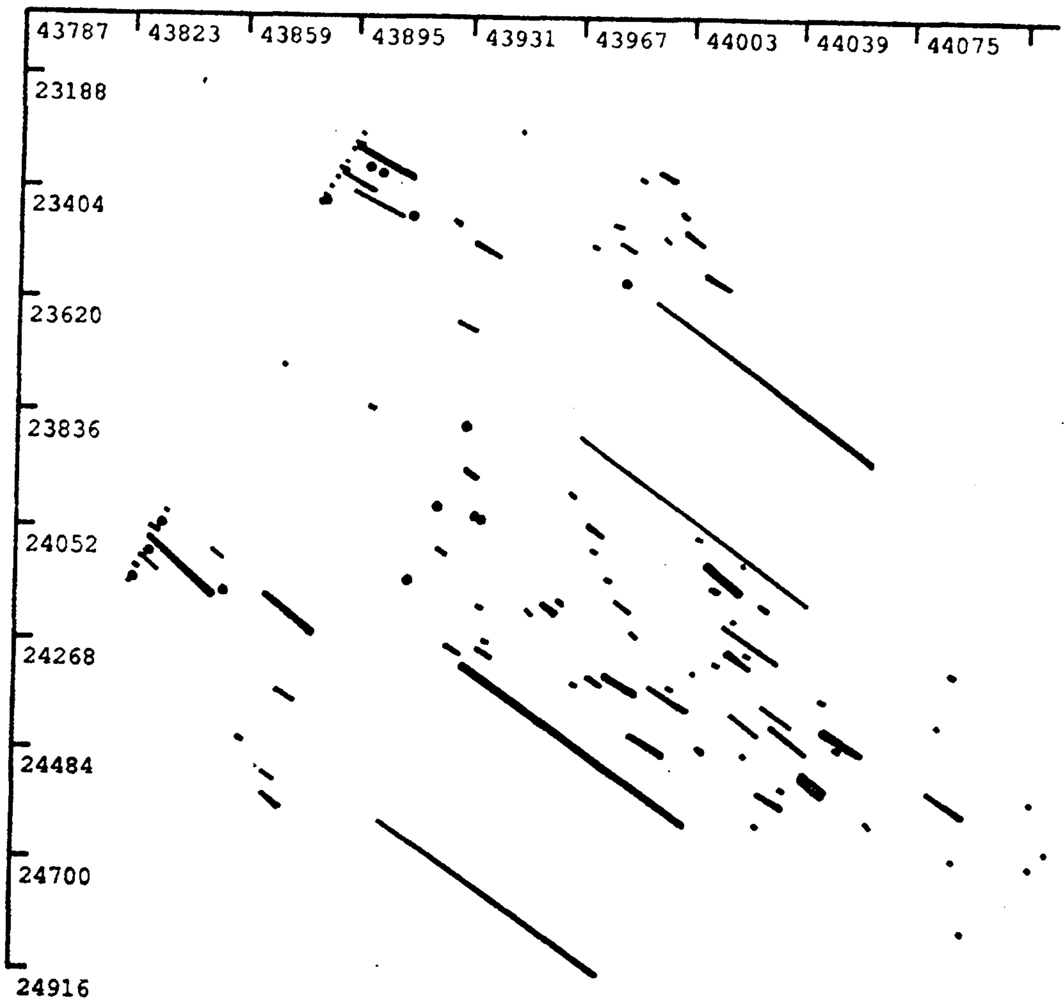

Figure A-36. Linear and online clusters from data set 10.

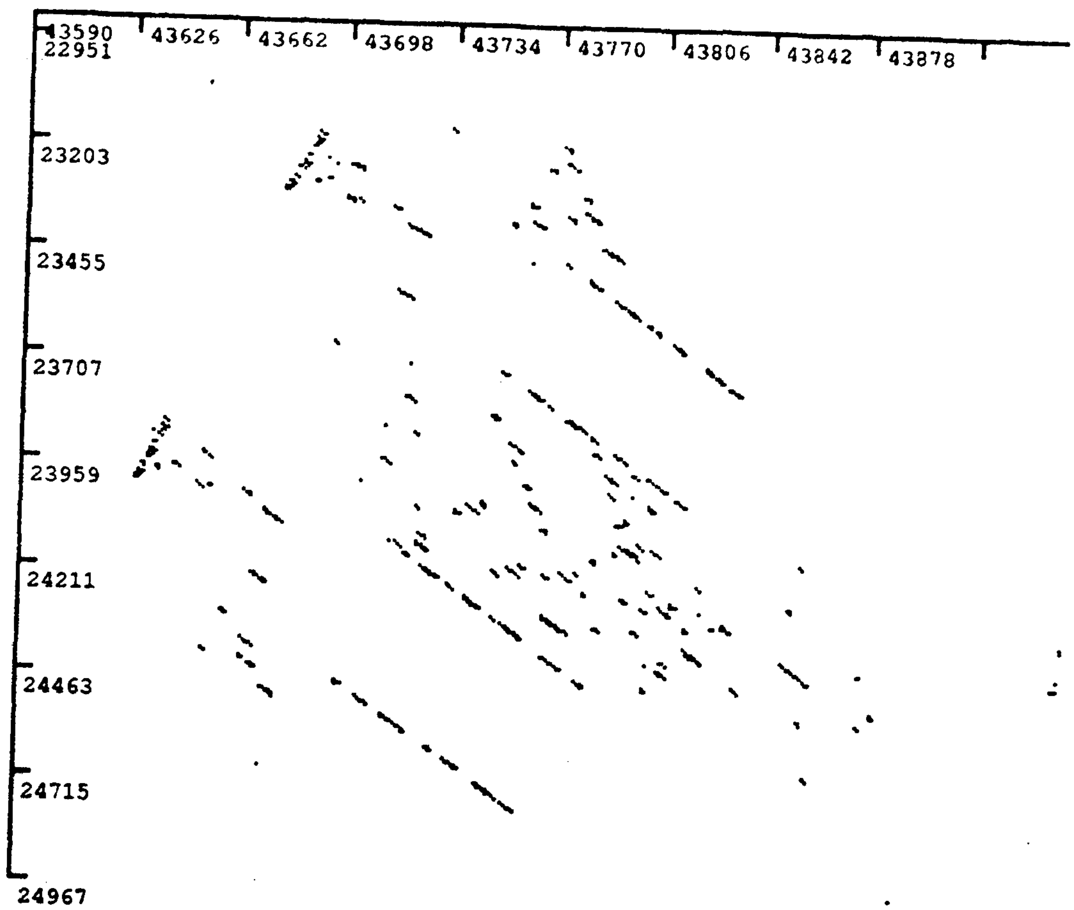

Figure A-37. Input data from data set 11.

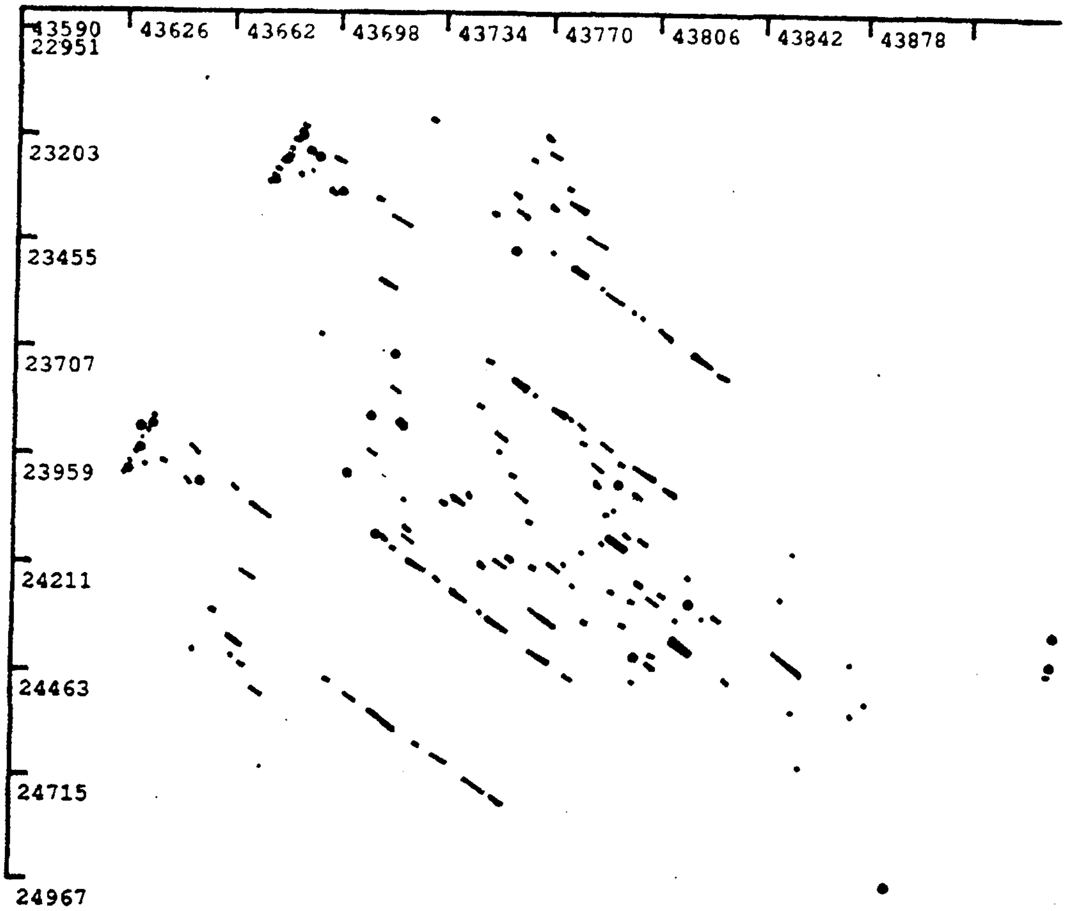

Figure A-38. Initial clusters from data set 11.

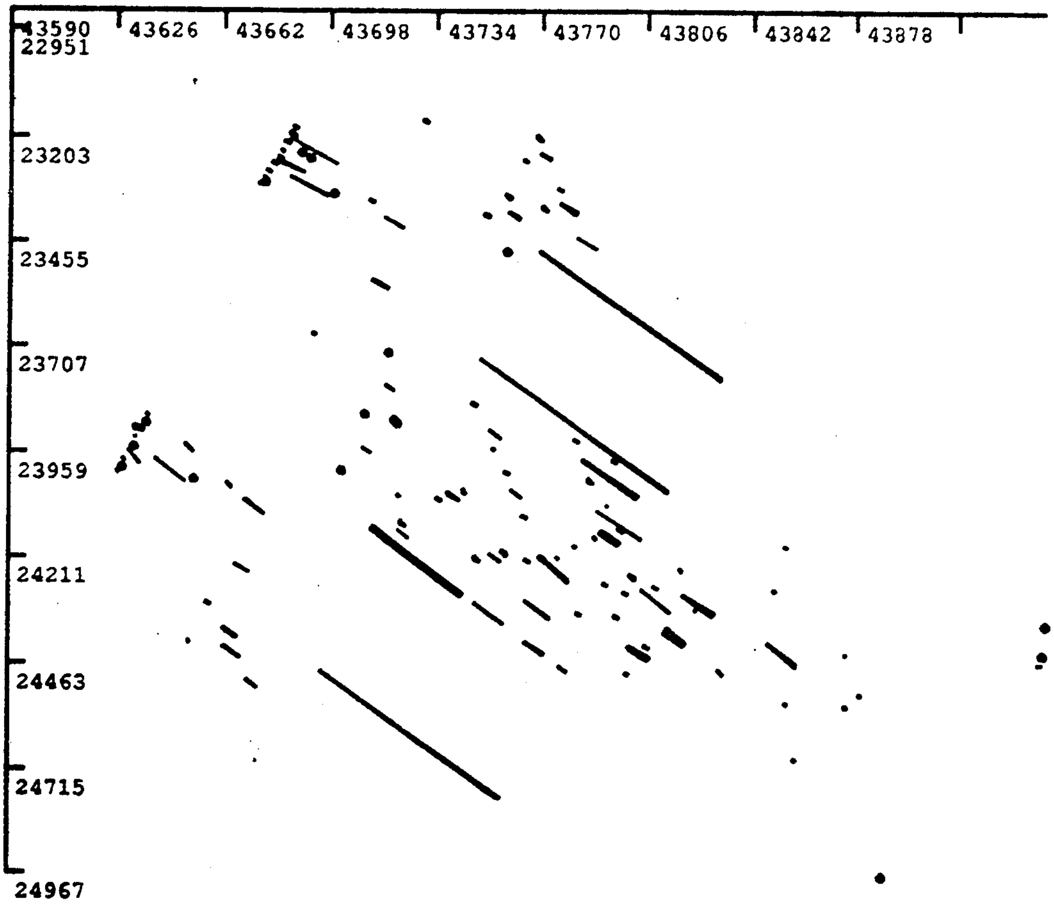

Figure A-39. Linear and online clusters from data set 11.

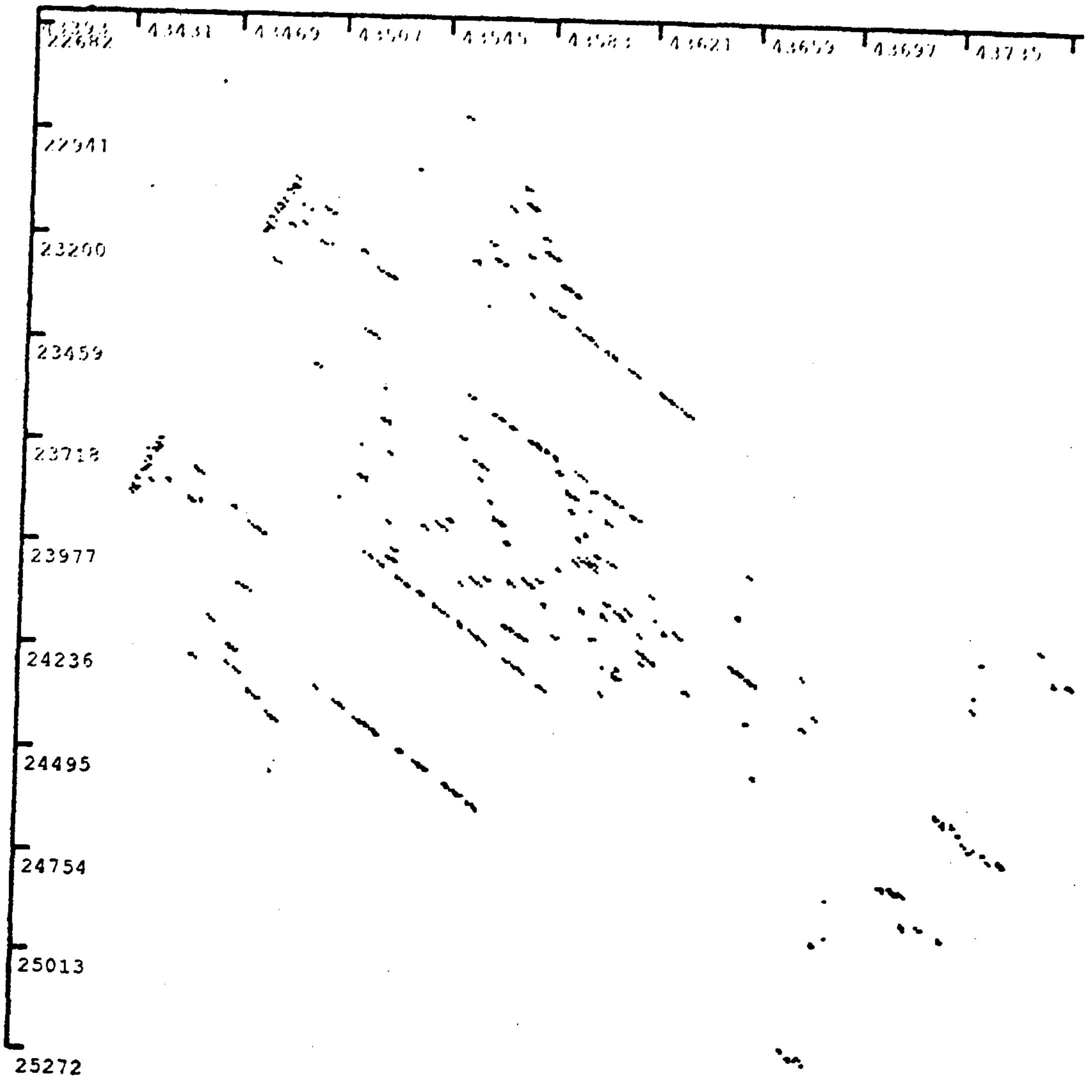

Figure A-40. Input data from data set 12.

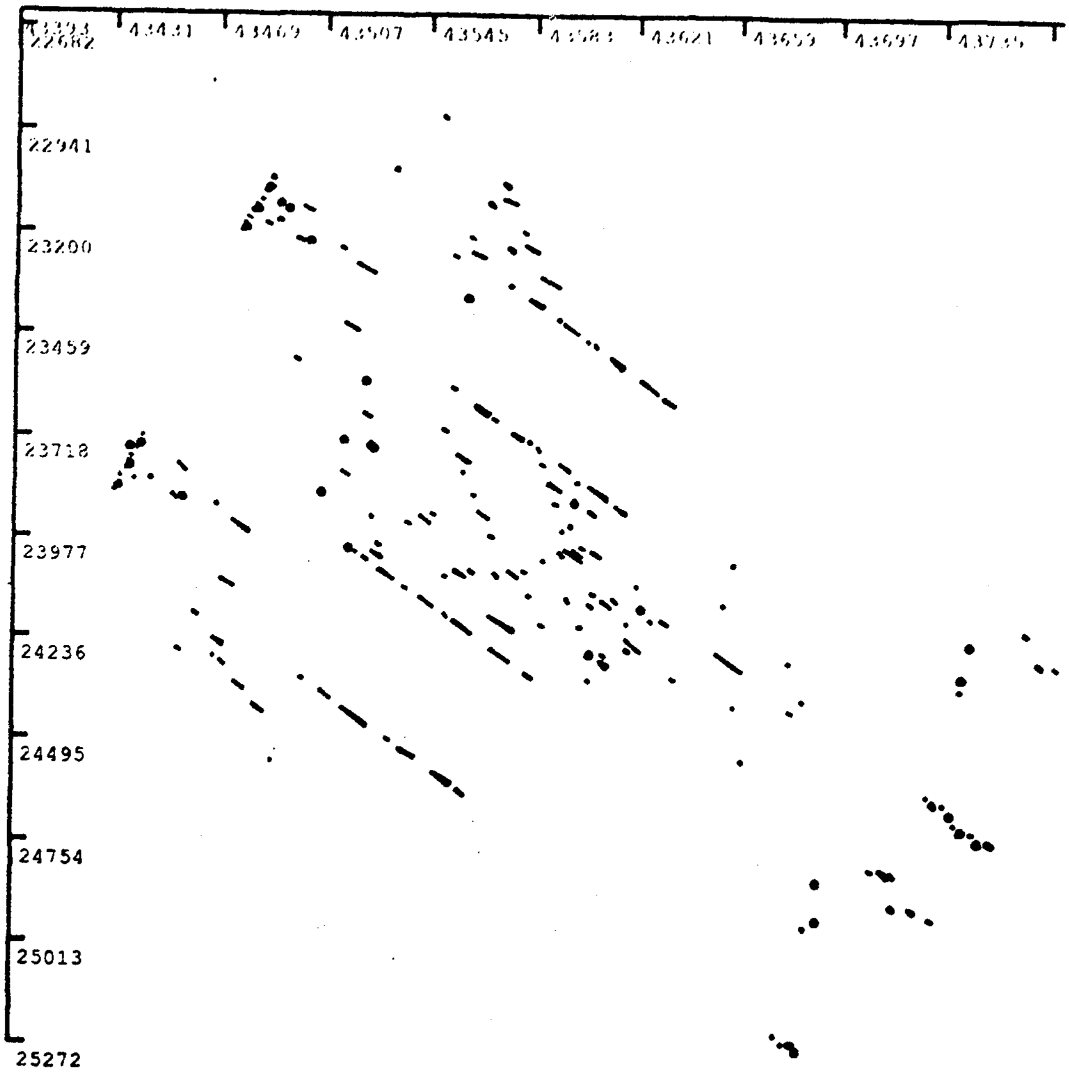

Figure A-41. Initial clusters from data set 12.

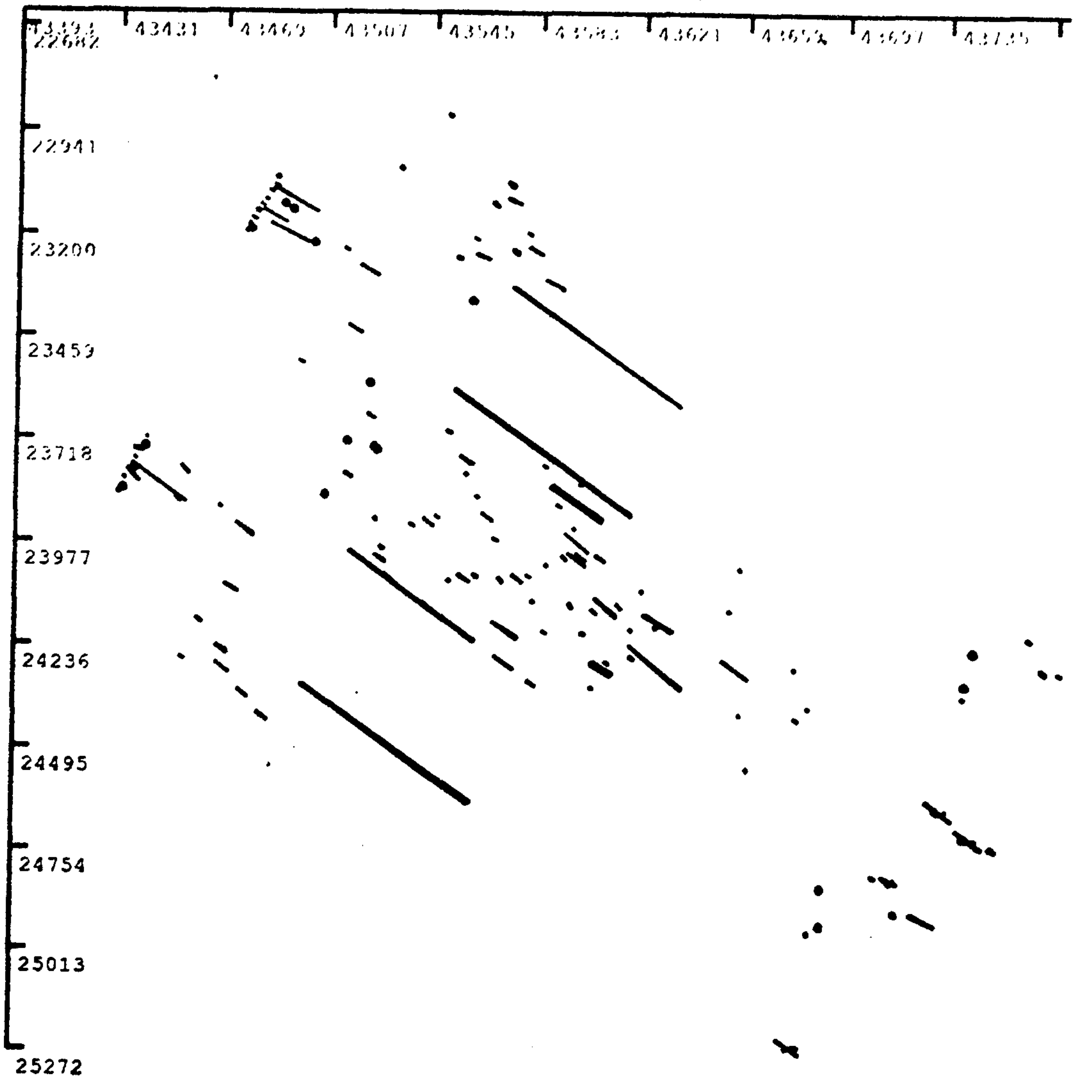

Figure A-42. Linear and online clusters from data set 12.

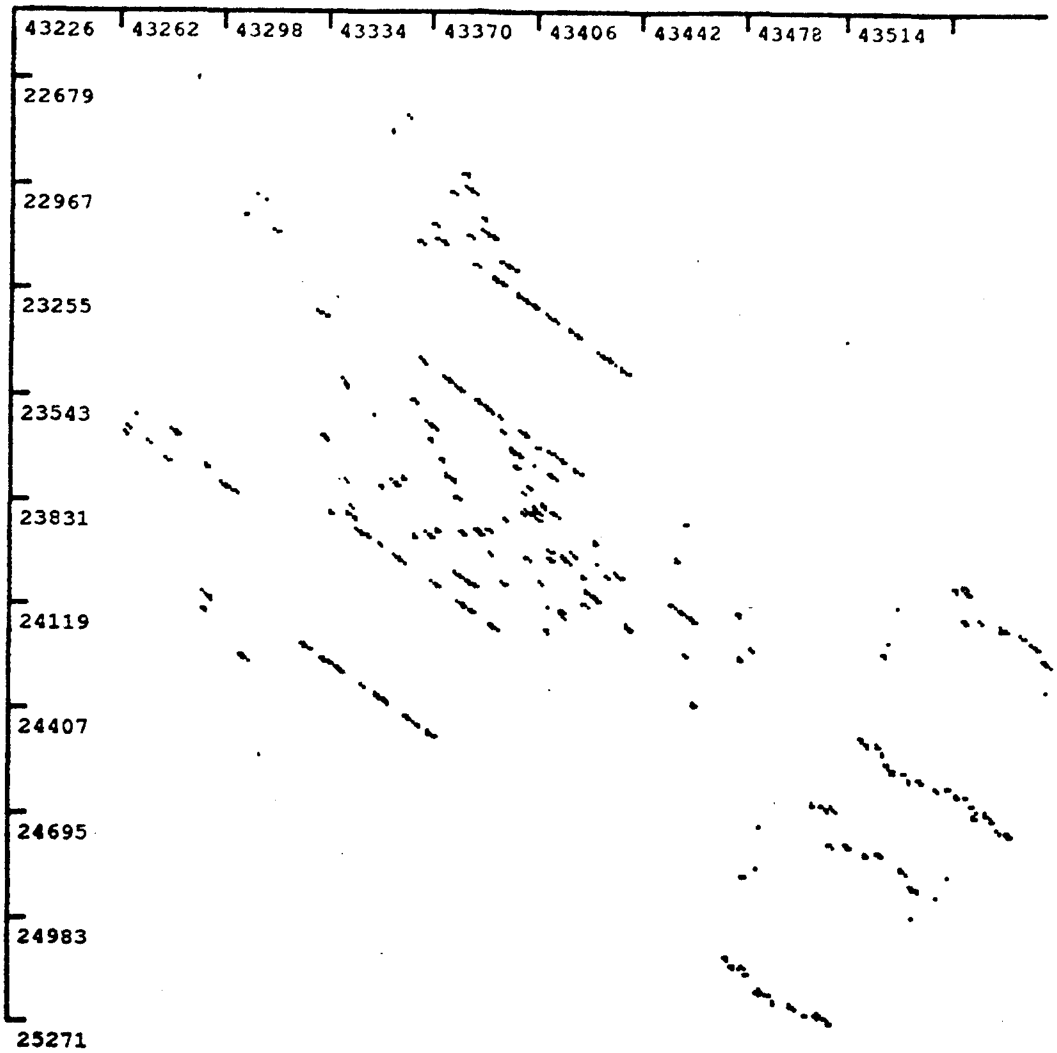

Figure A-43. Input data from data set 13.

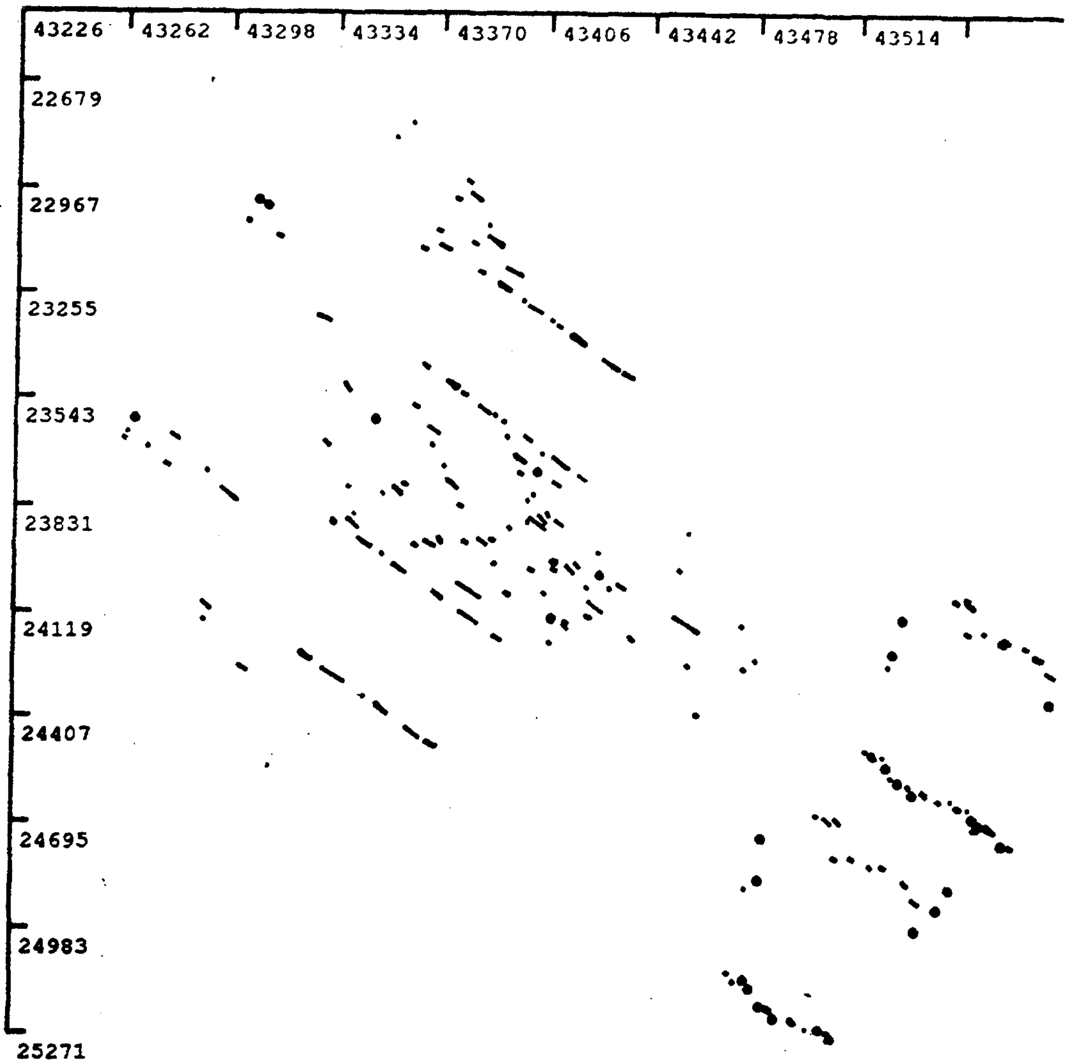

Figure A-44. Initial clusters from data set 13.

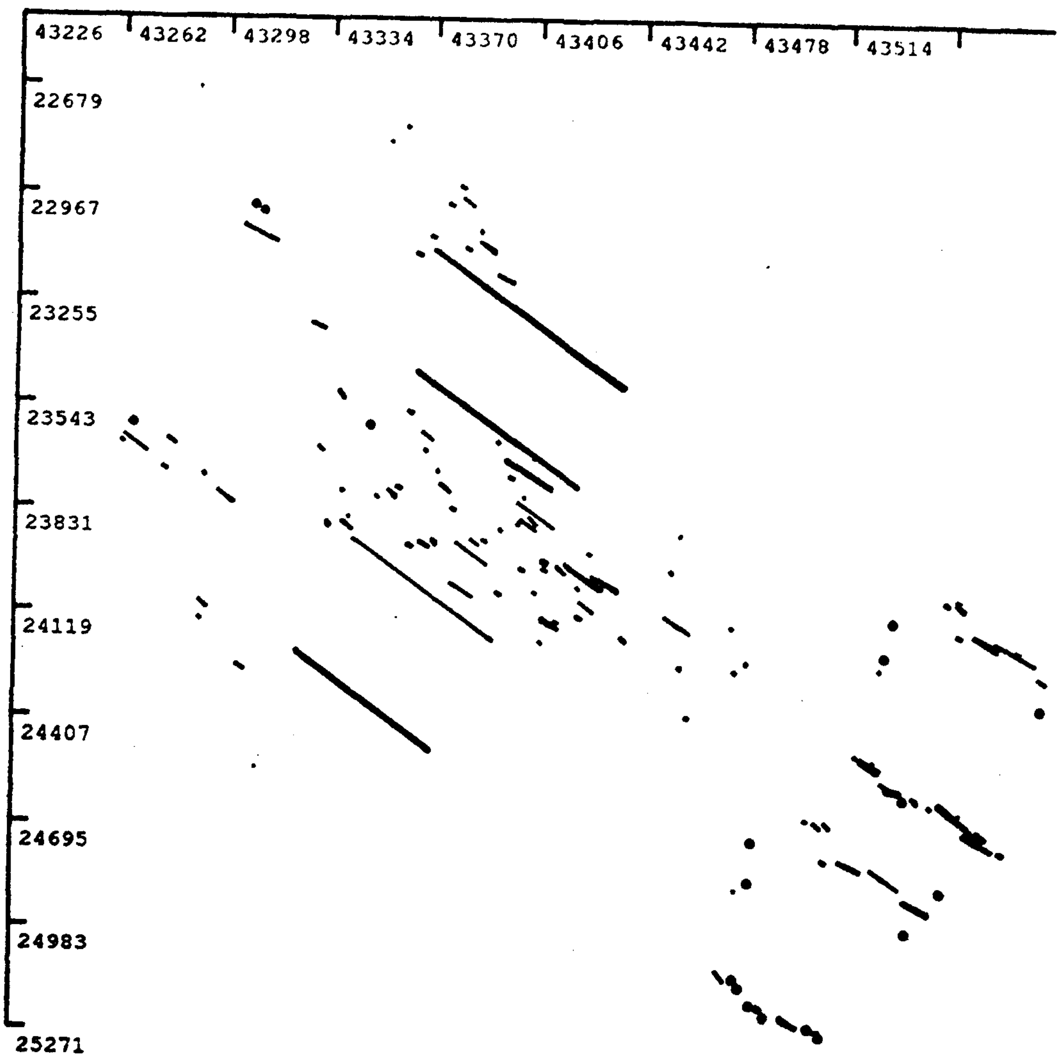

Figure A-45. Linear and online clusters from data set 13.

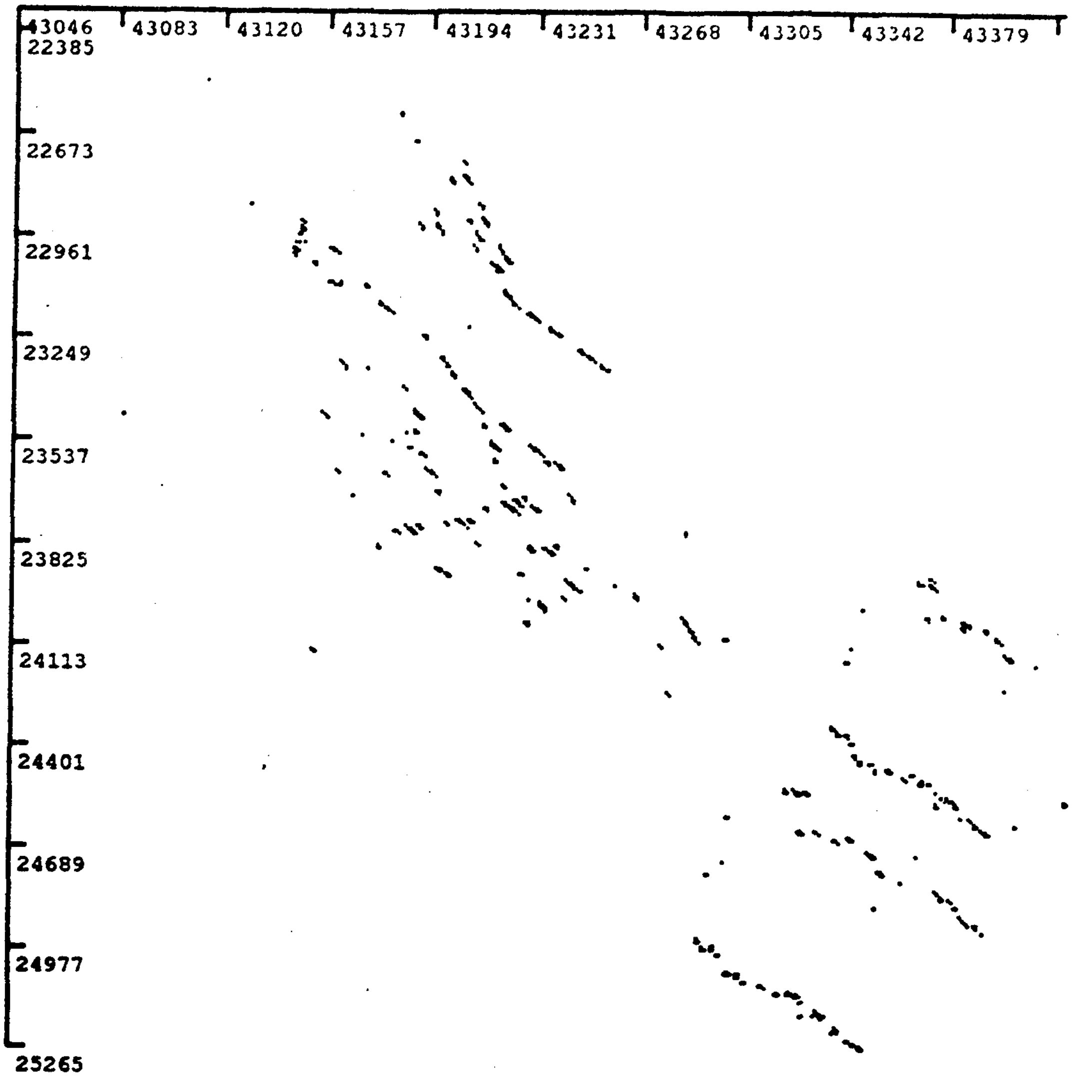

Figure A-46. Input data from data set 14.

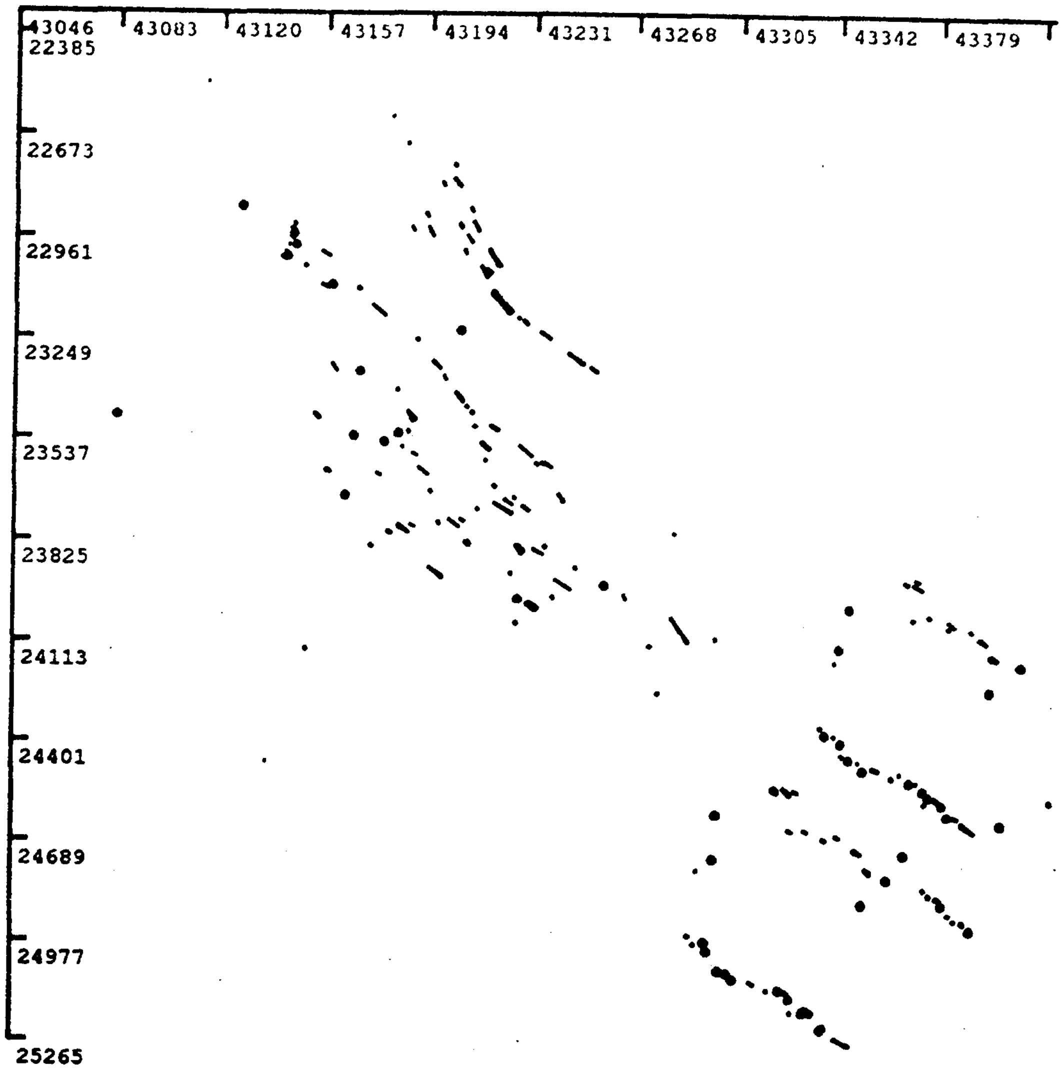

Figure A-47. Initial clusters from data set 14.

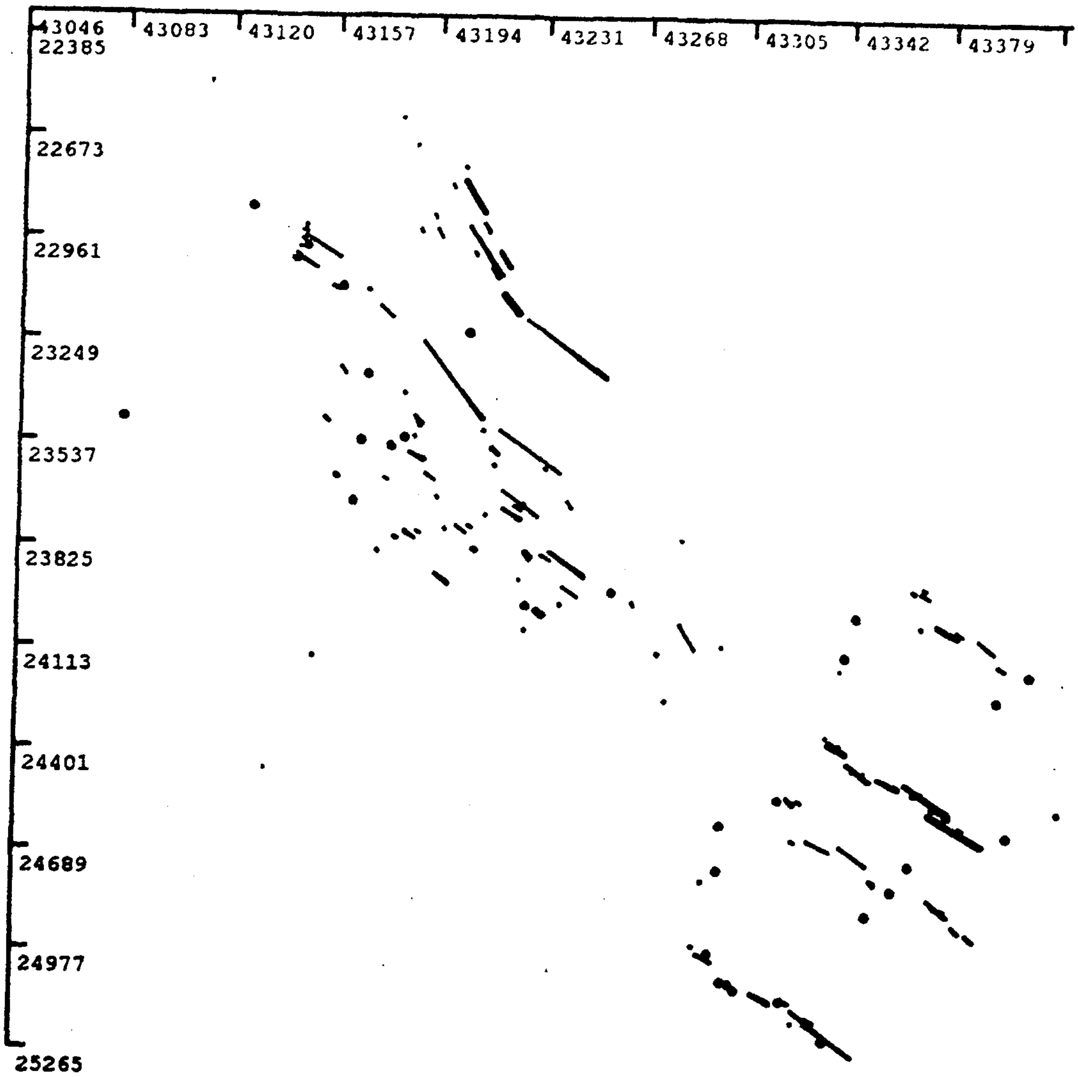

Figure A-48. Linear and online clusters from data set 14.

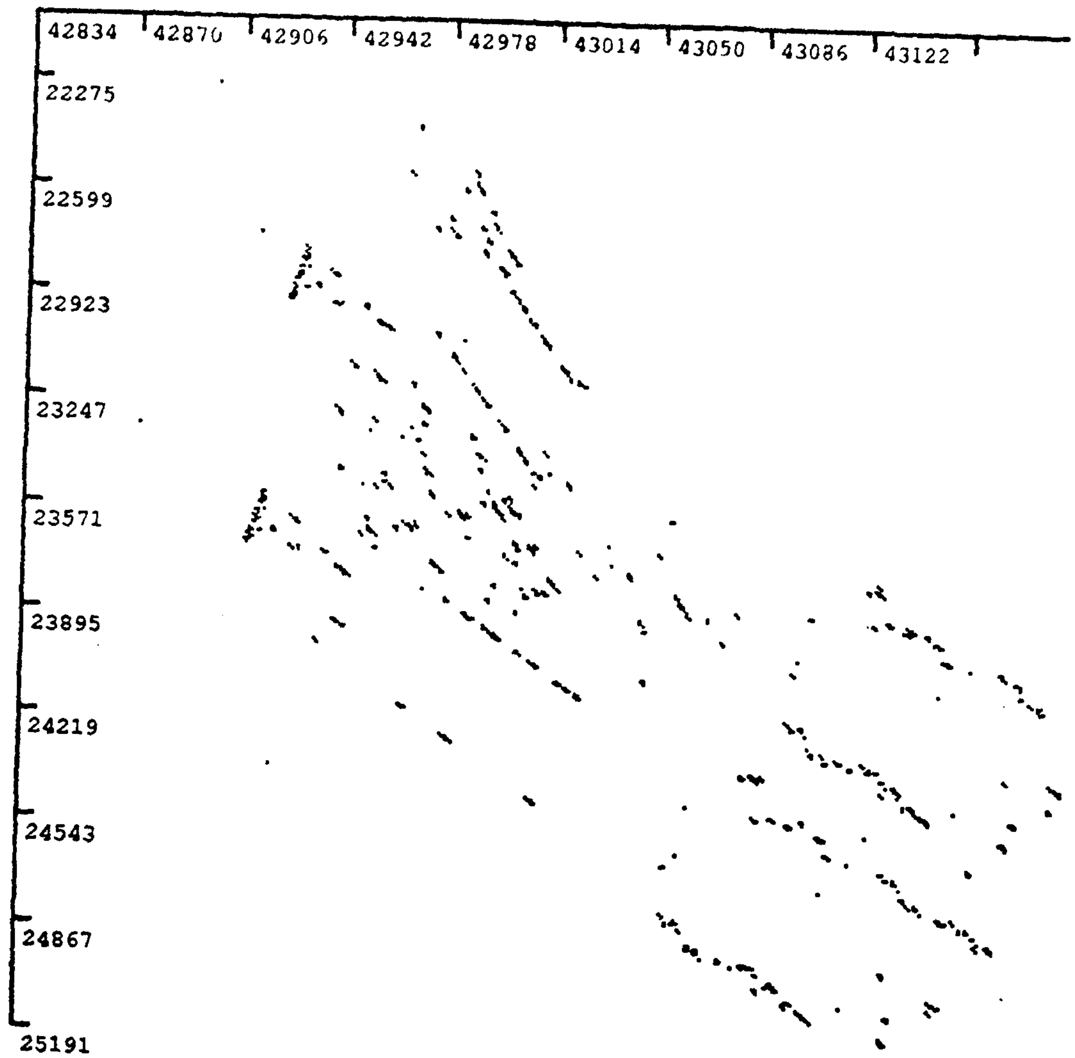

Figure A-49. Input data from data set 15.

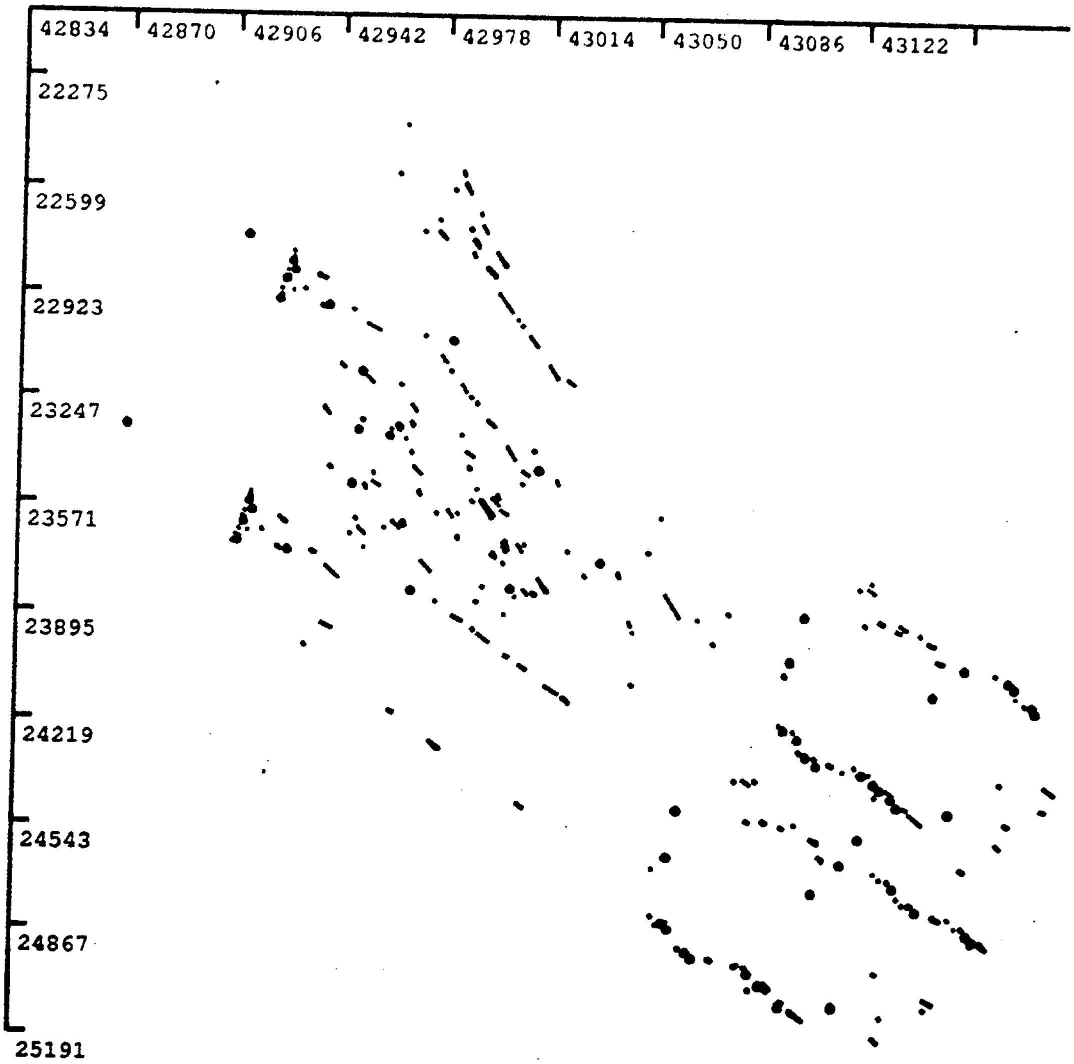

Figure A-50. Initial clusters from data set 15.

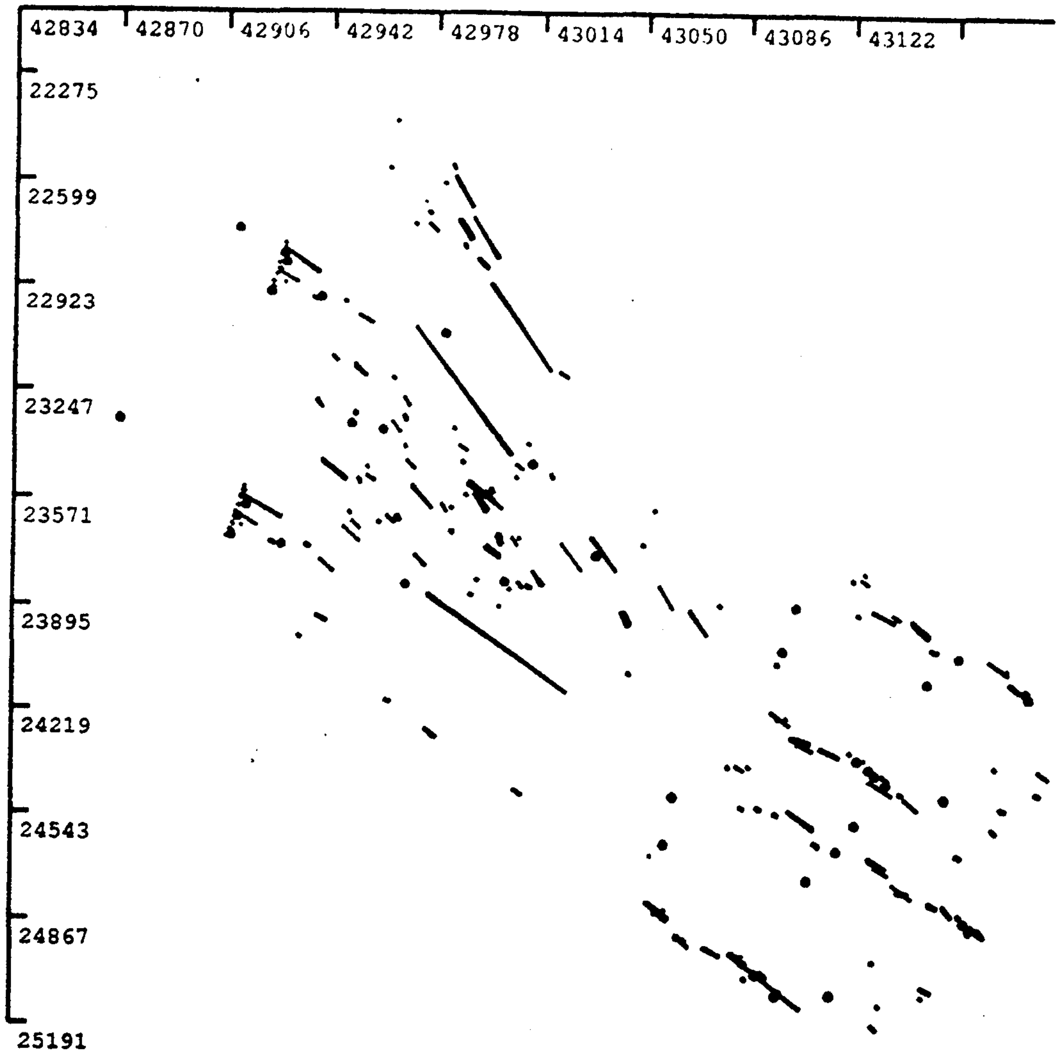

Figure A-51. Linear and online clusters from data set 15.

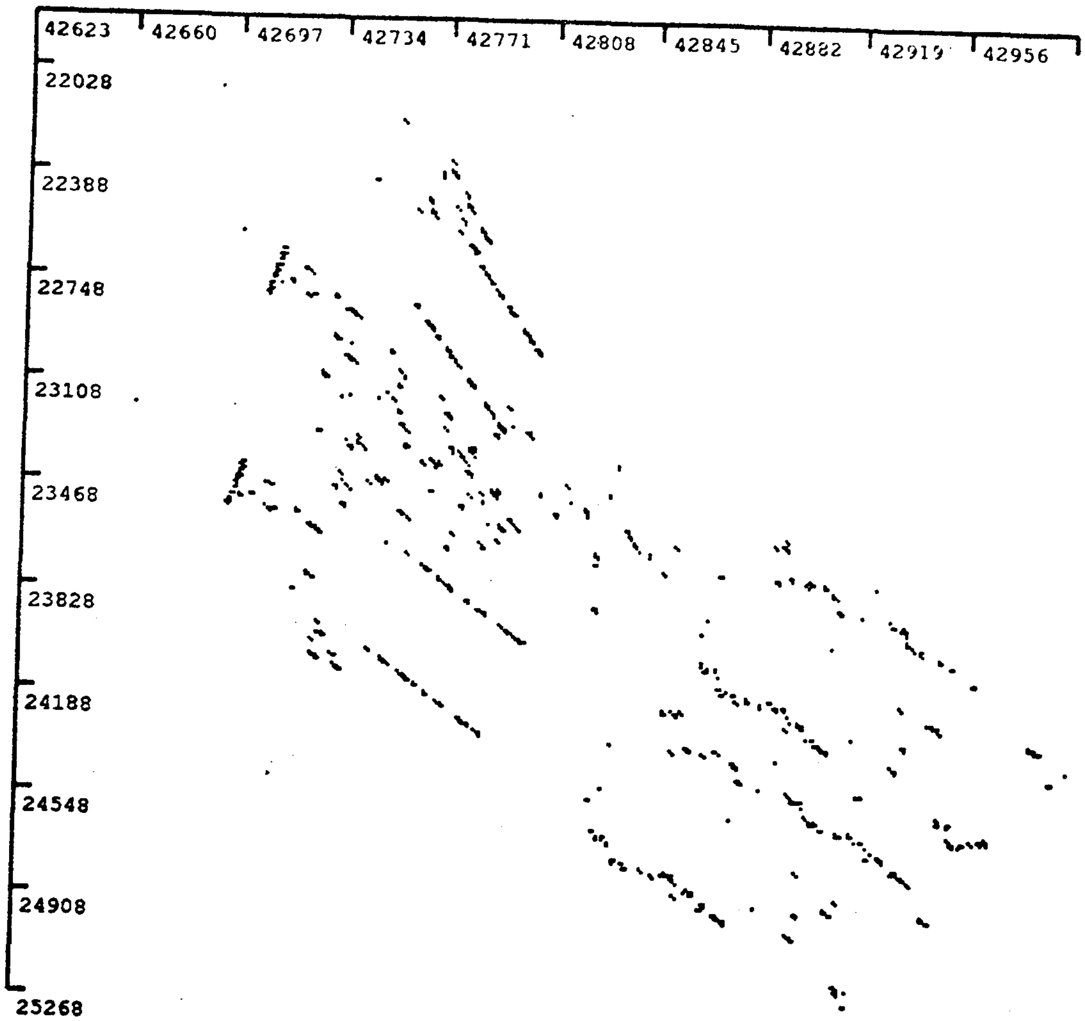

Figure A-52. Input data from data set 16.

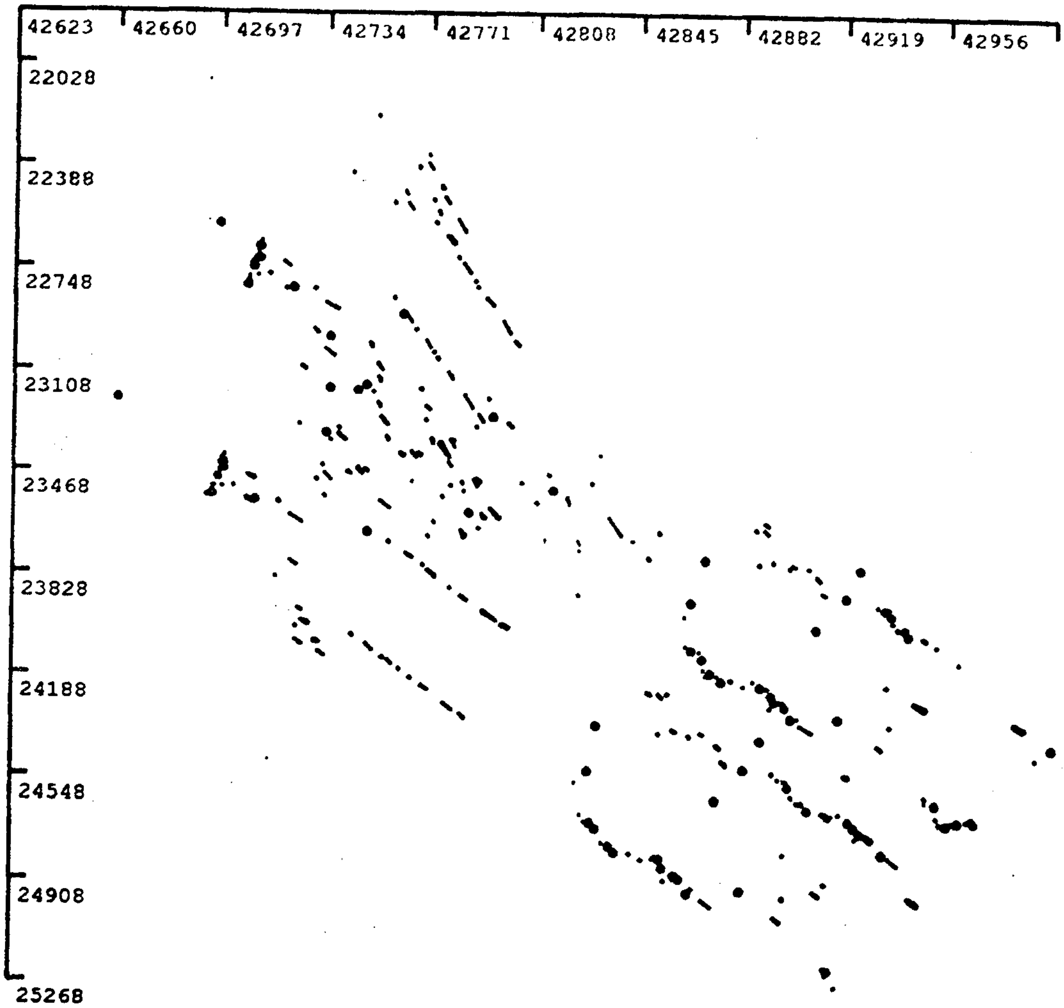

Figure A-53. Initial clusters from data set 16.

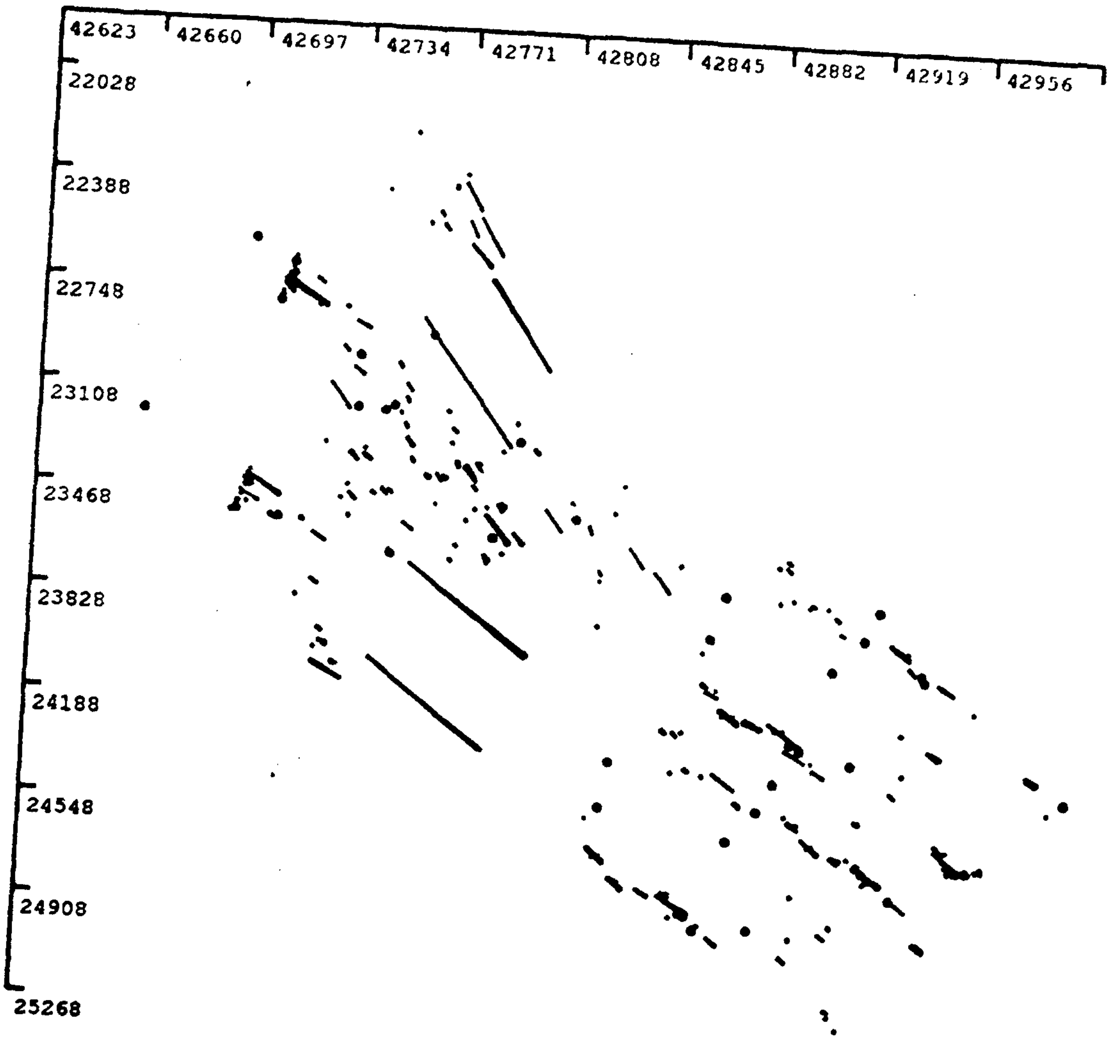

Figure A-54. Linear and online clusters from data set 16.

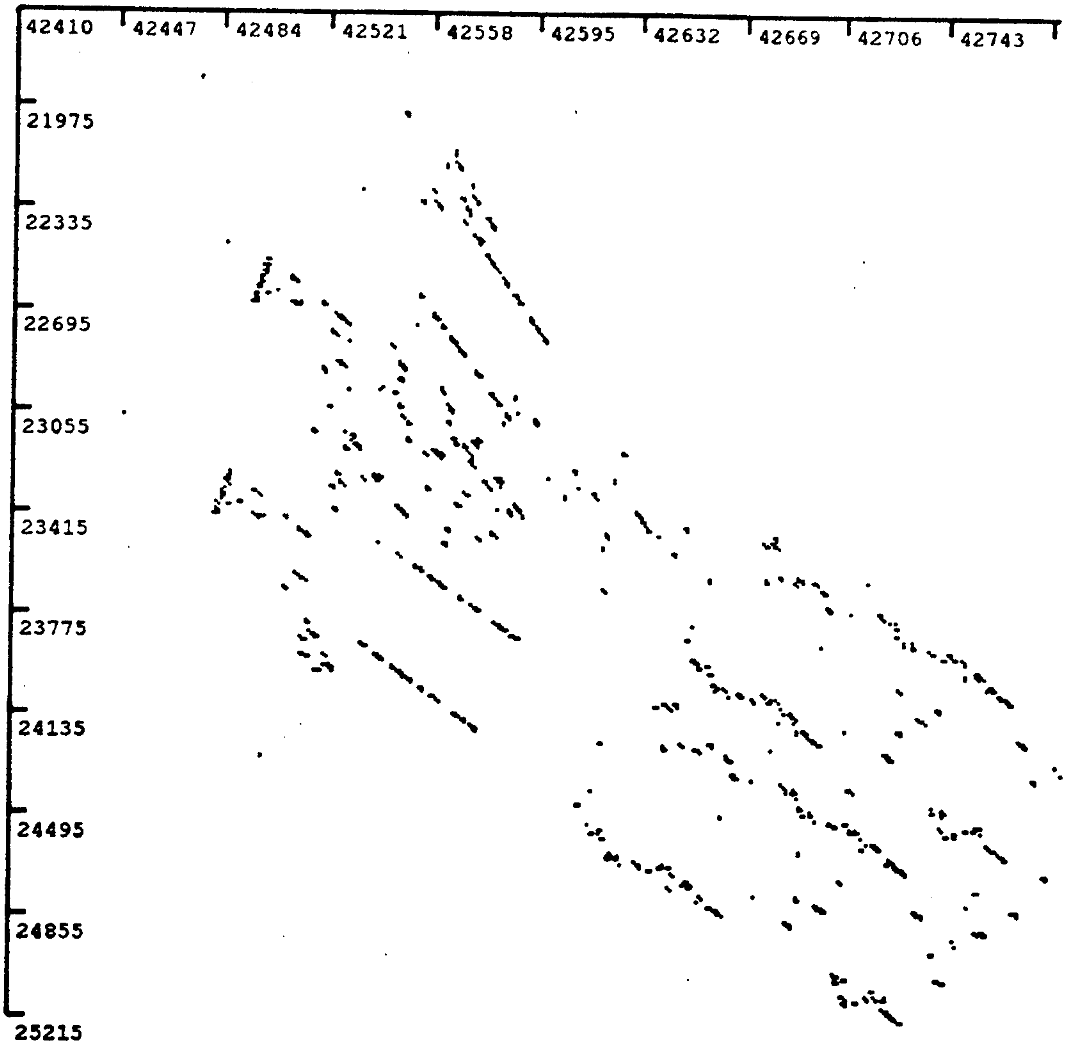

Figure A-55. Input data from data set 17.

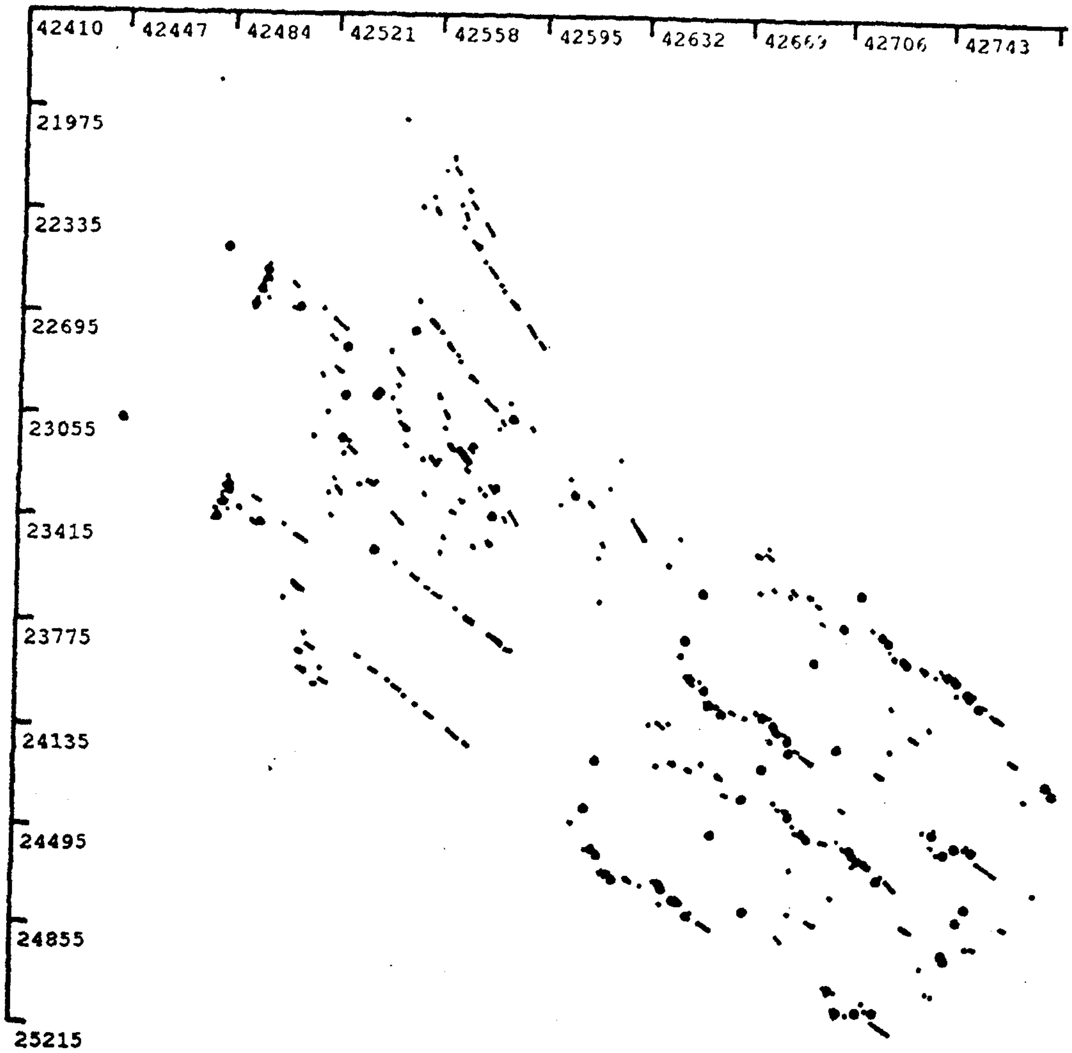

Figure A-56. Initial clusters from data set 17.

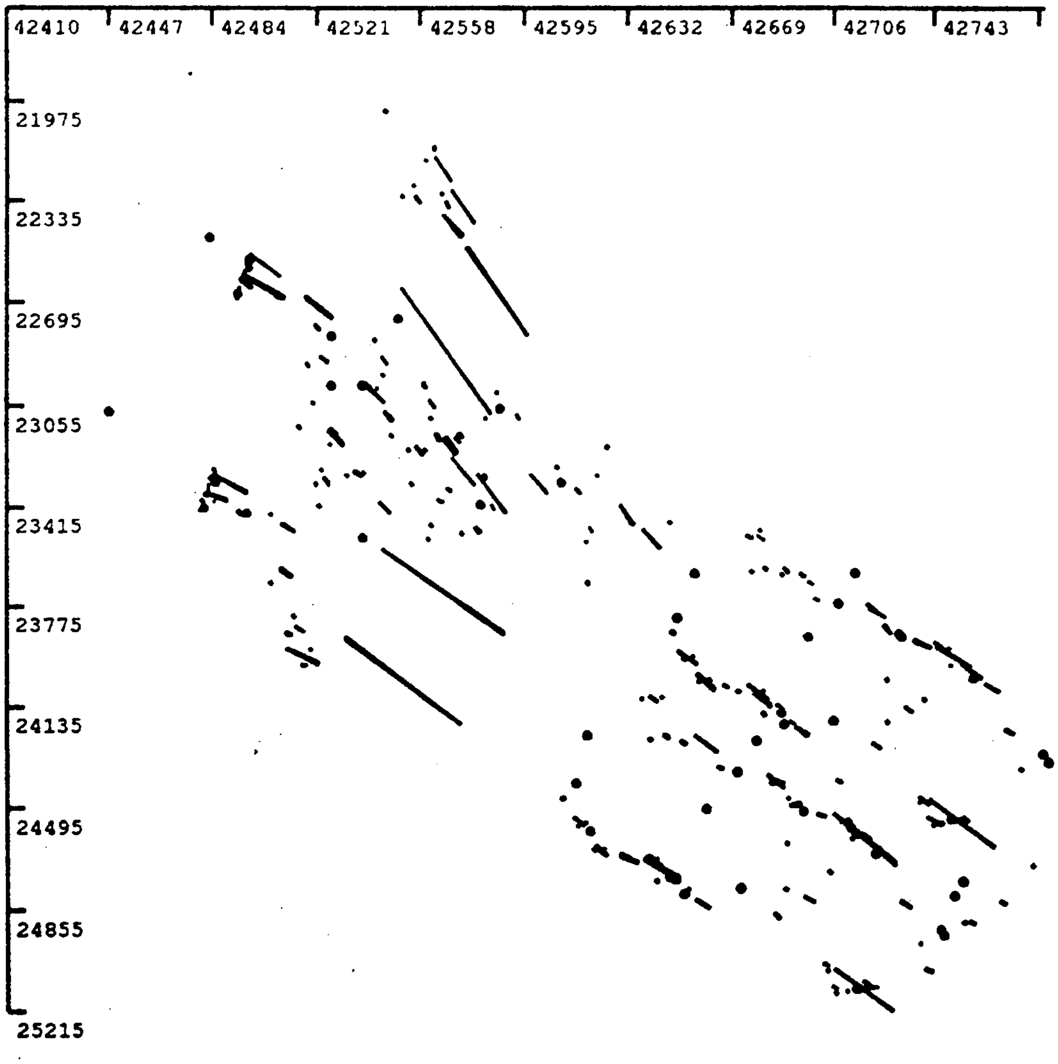

Figure A-57. Linear and online clusters from data set 17.

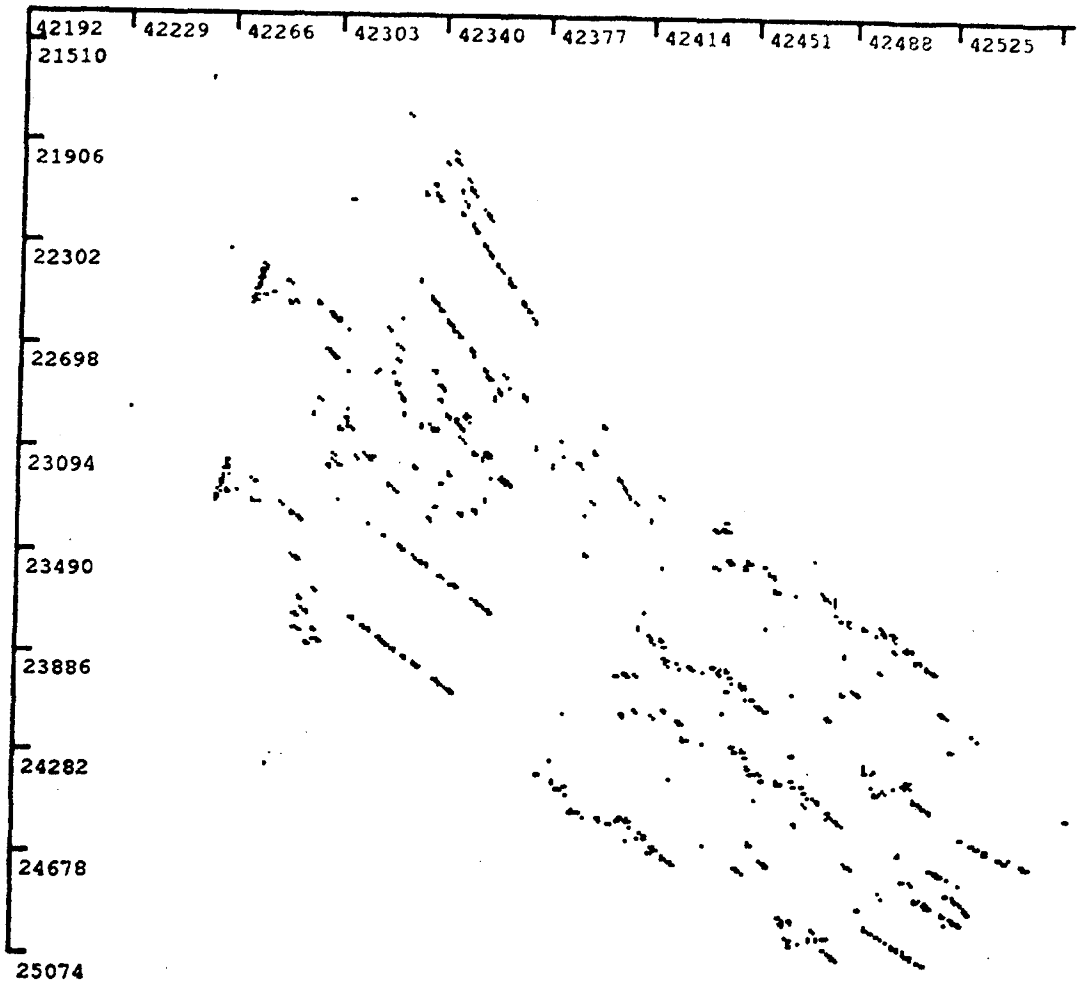

Figure A-58. Input data from data set 18.

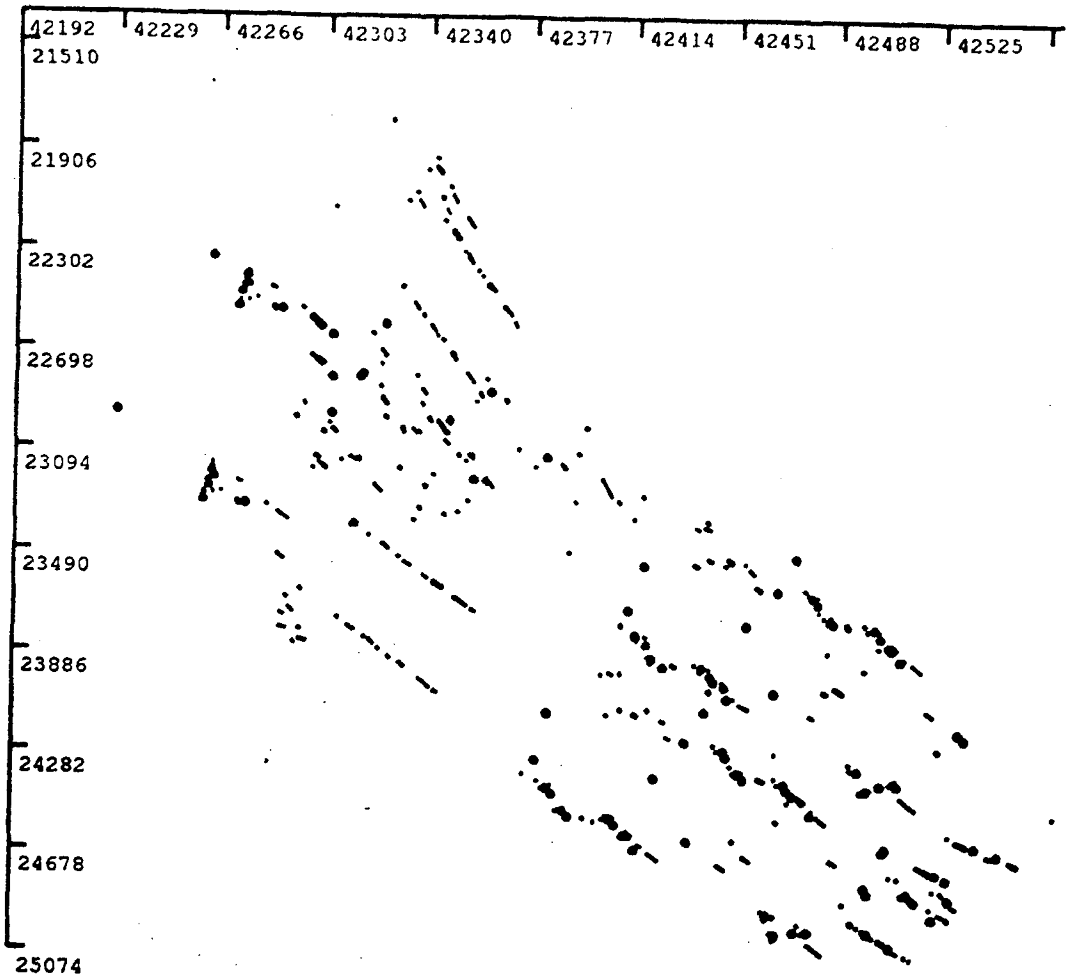

Figure A-59. Initial clusters from data set 18.

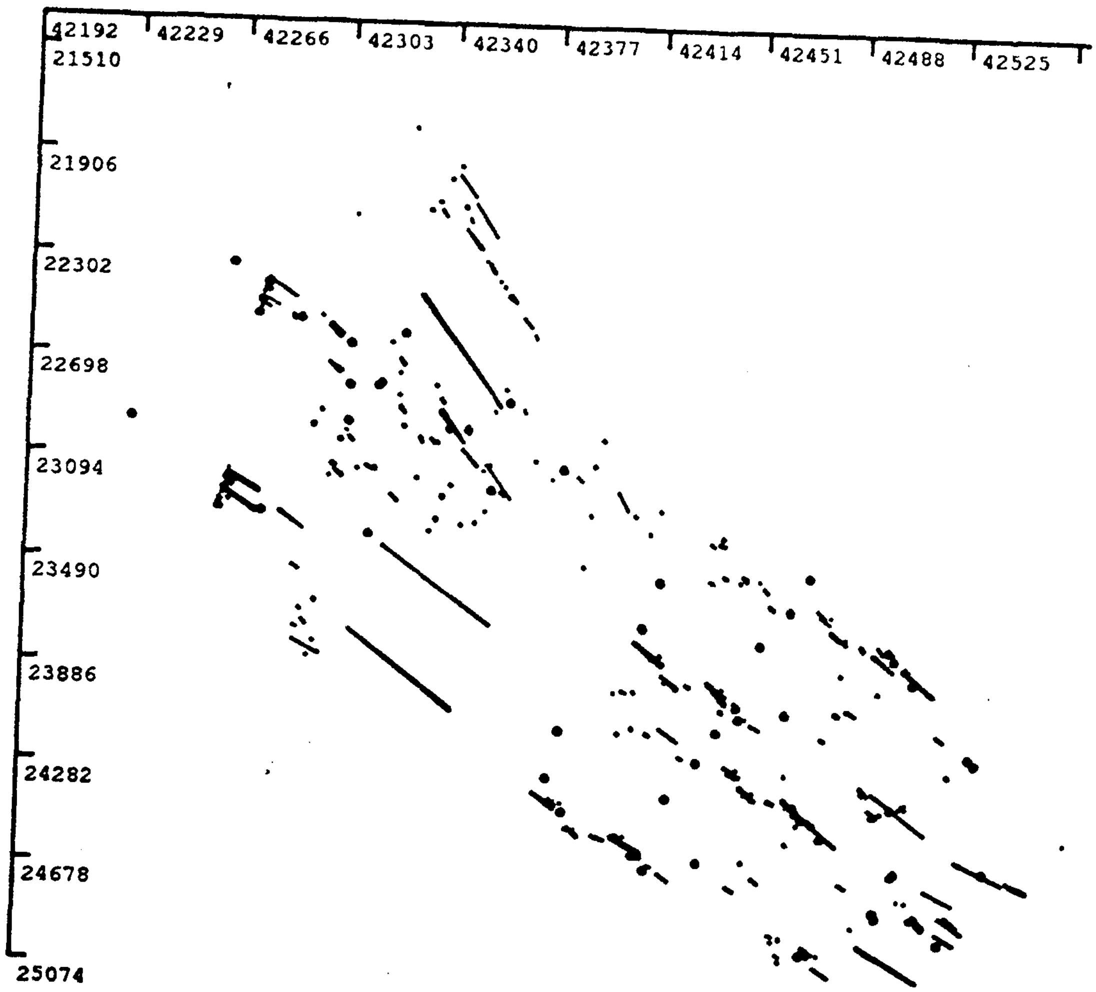

Figure A-60. Linear and online clusters from data set 18.

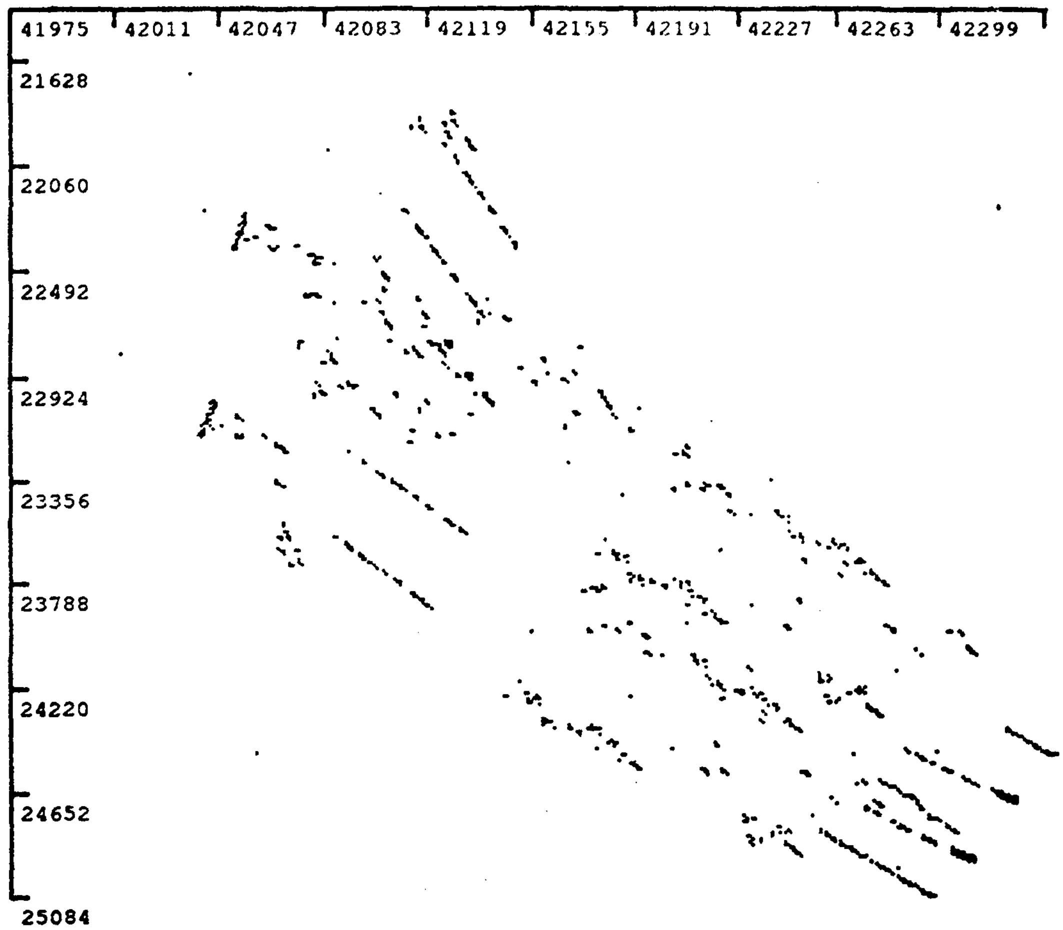

Figure A-61. Input data from data set 19.

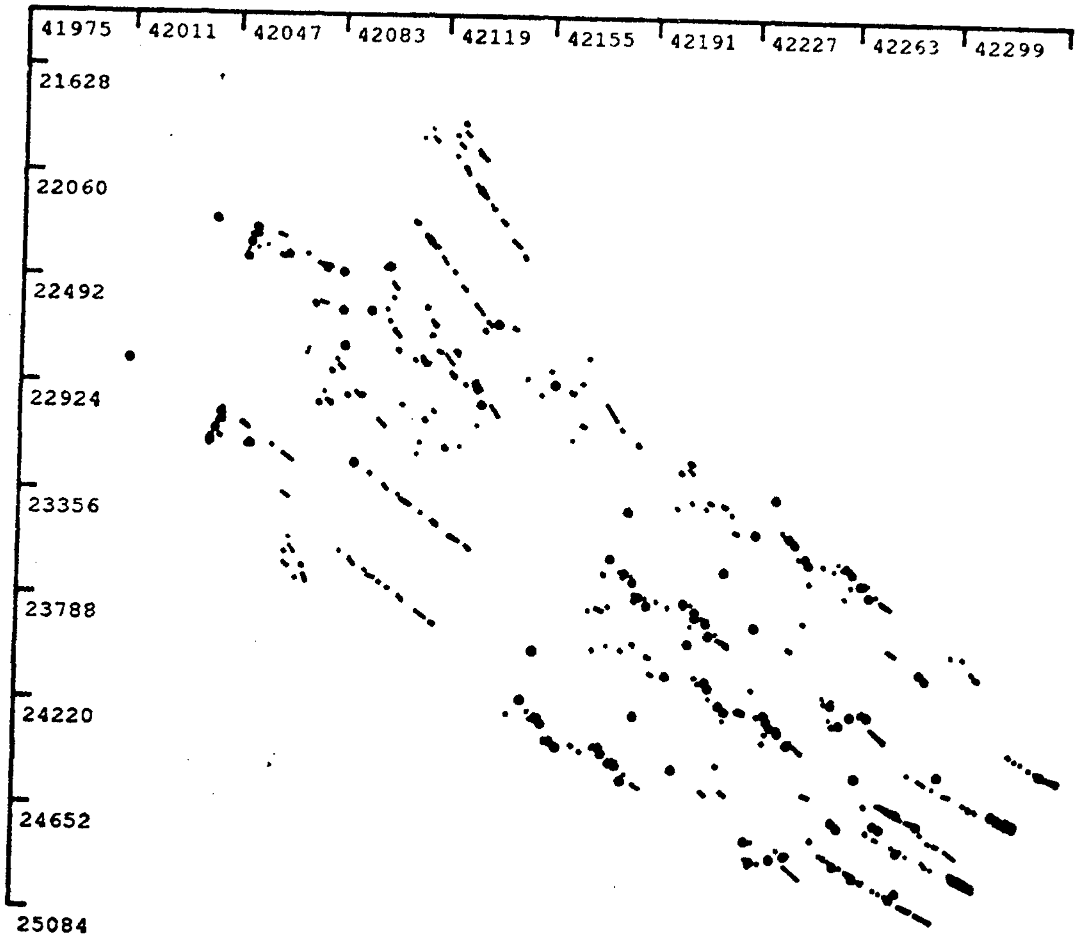

Figure A-62. Initial clusters from data set 19.

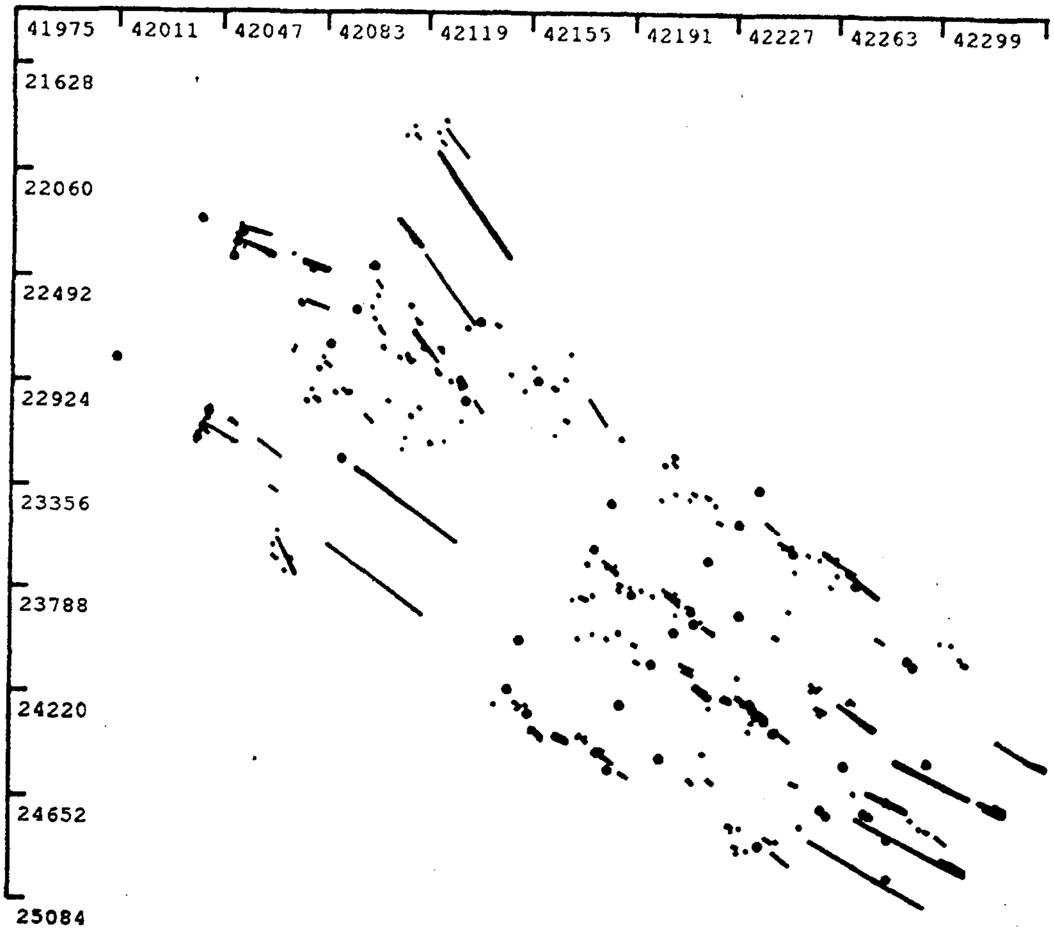

Figure A-63. Linear and online clusters from data set 19.

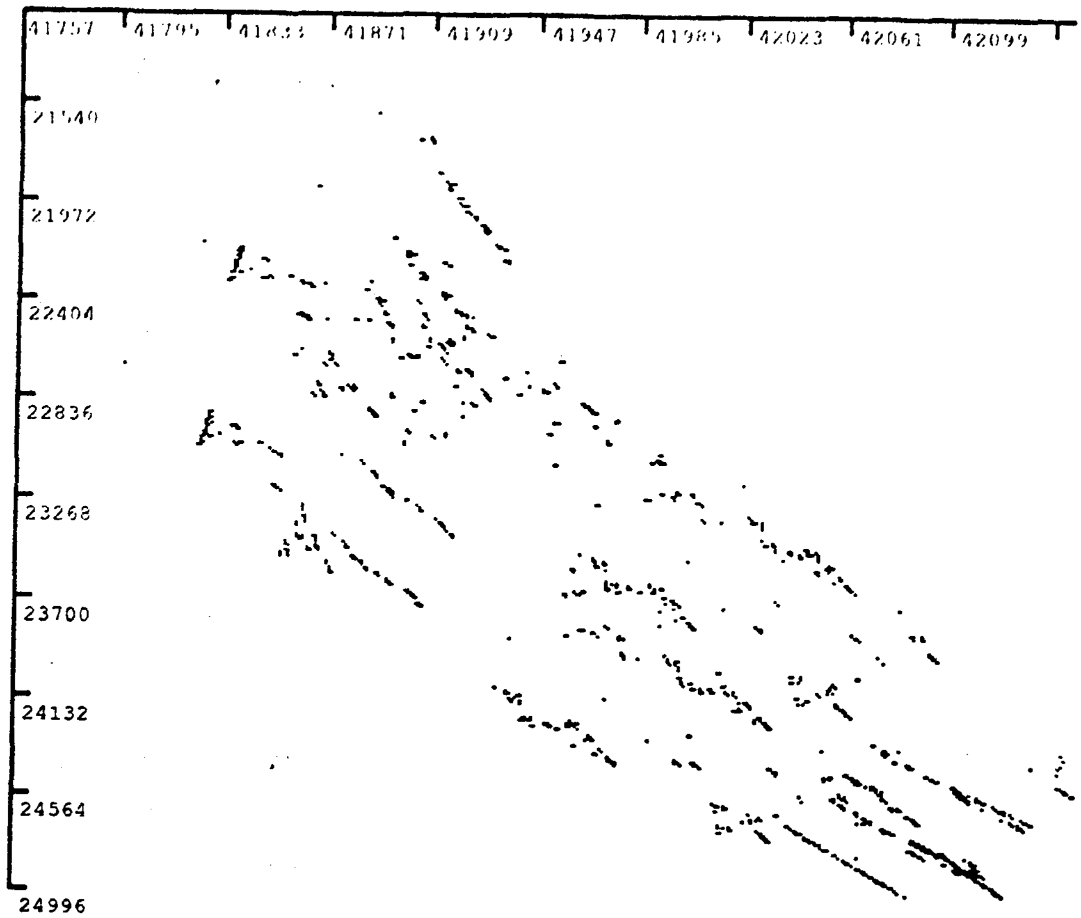

Figure A-64. Input data from data set 20.

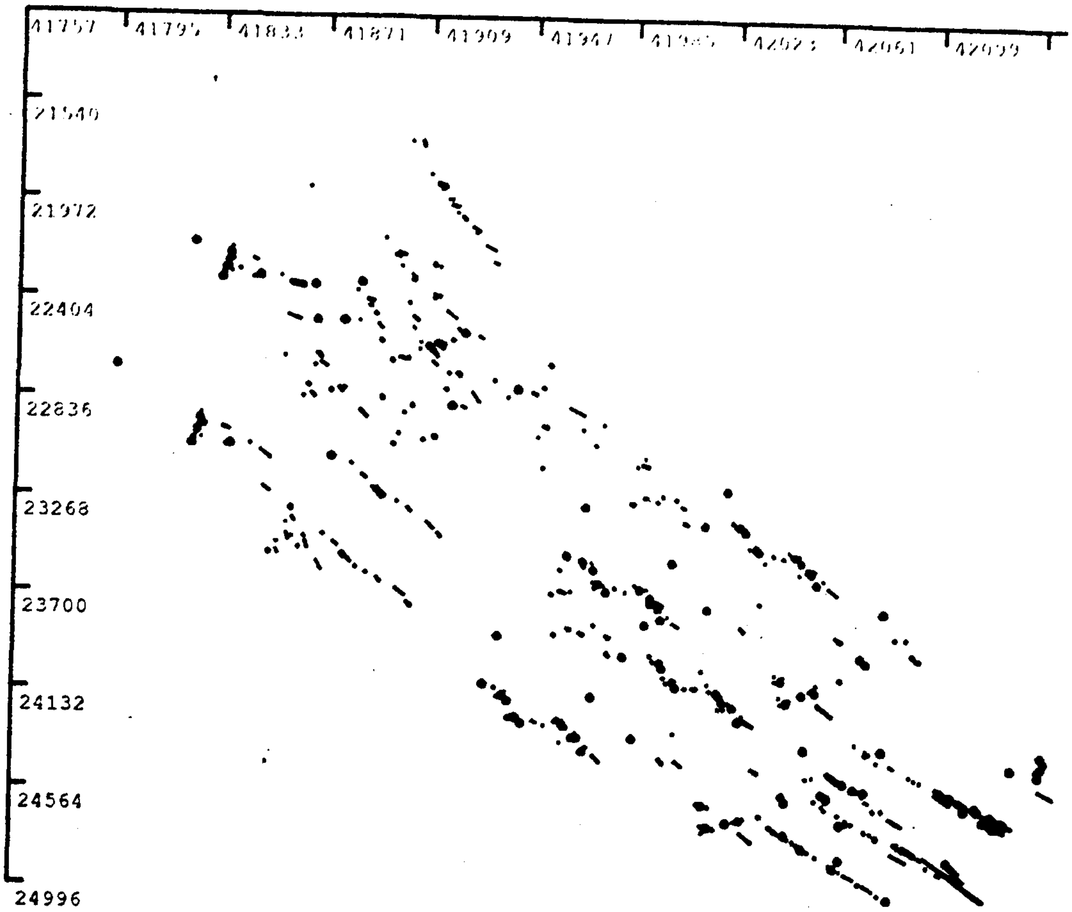

Figure A-65. Initial clusters from data set 20.

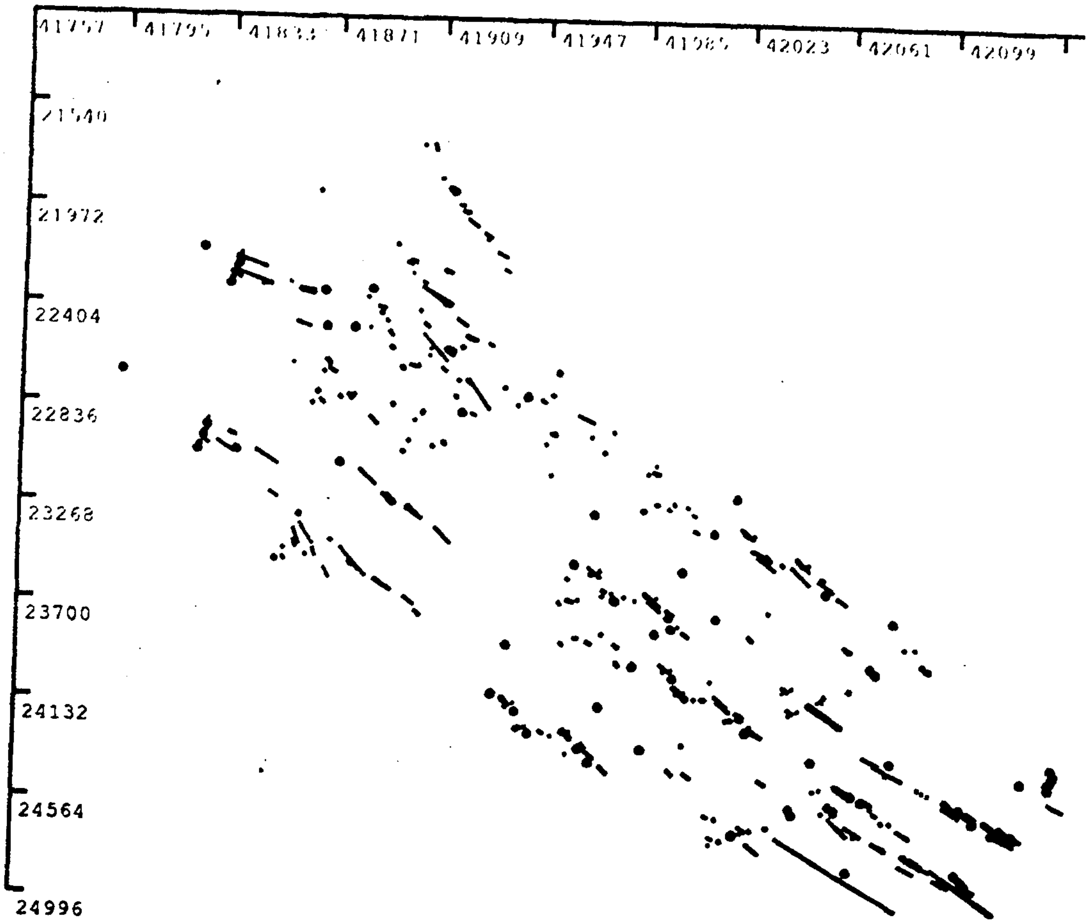

Figure A-66. Linear and online clusters from data set 20.

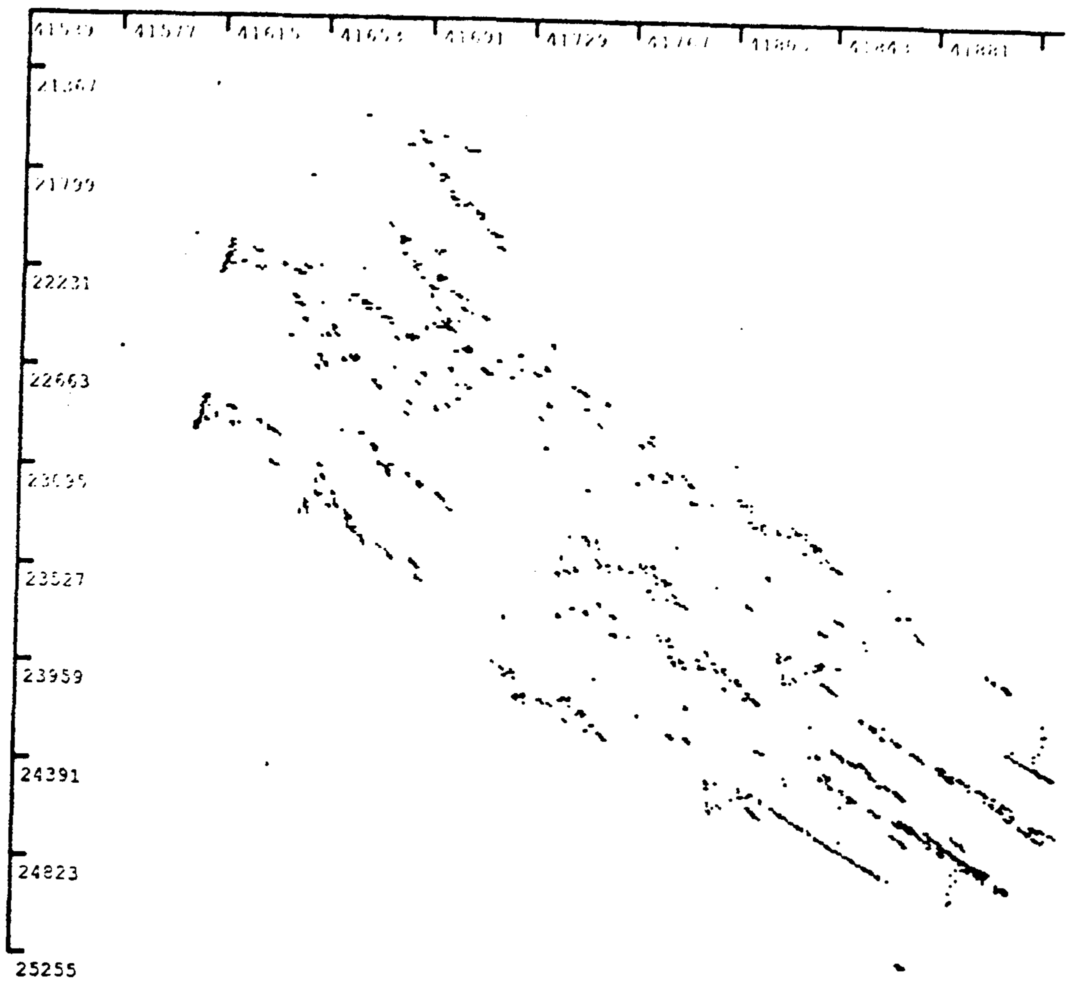

Figure A-67. Input data from data set 21.

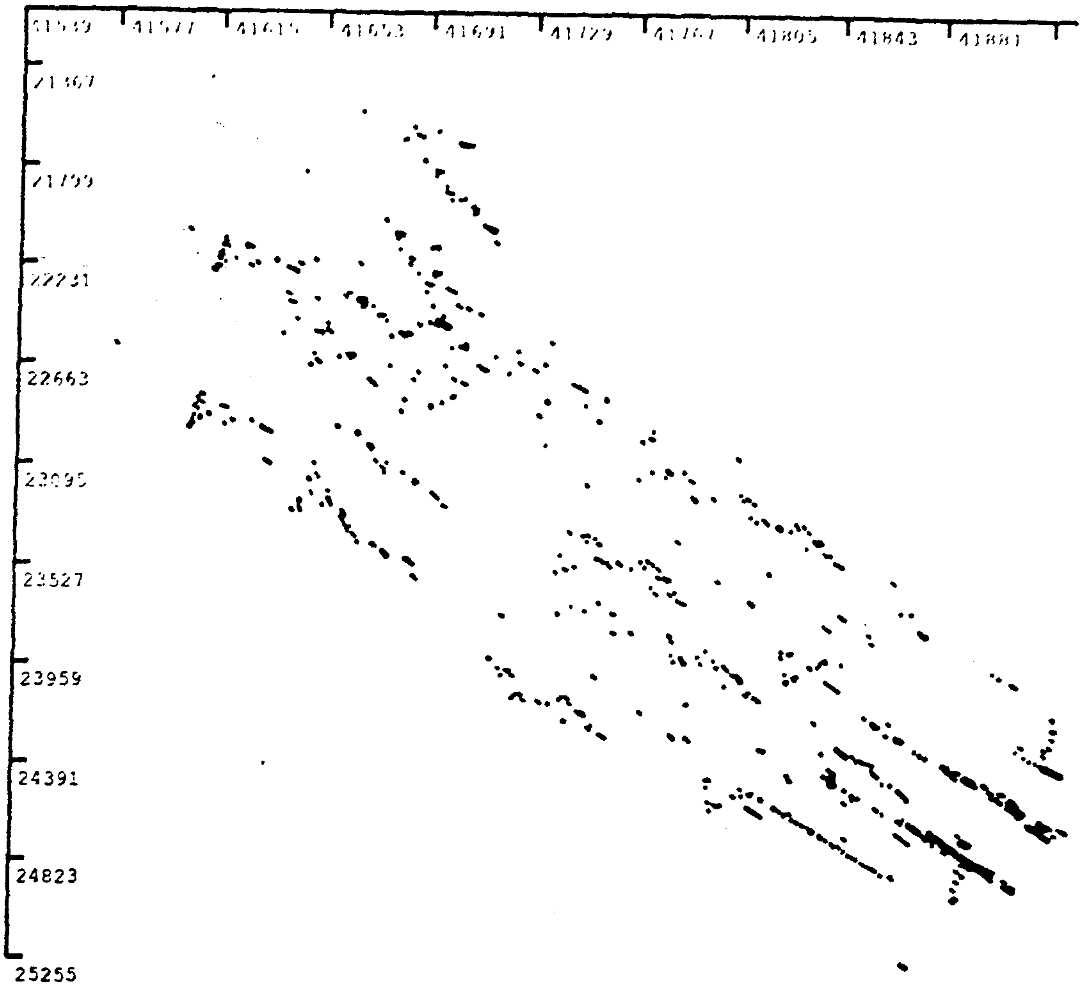

Figure A-68. Initial clusters from data set 21.

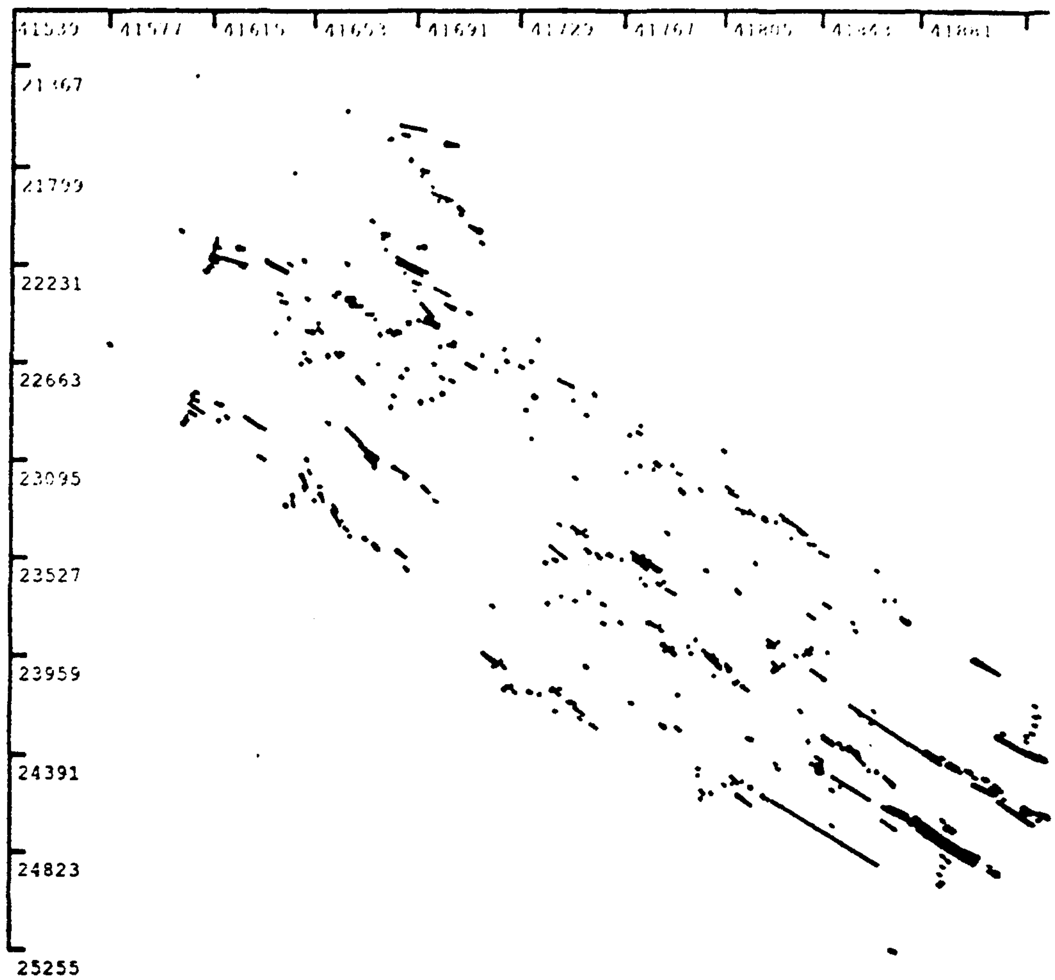

Figure A-69. Linear and online clusters from data set 21.

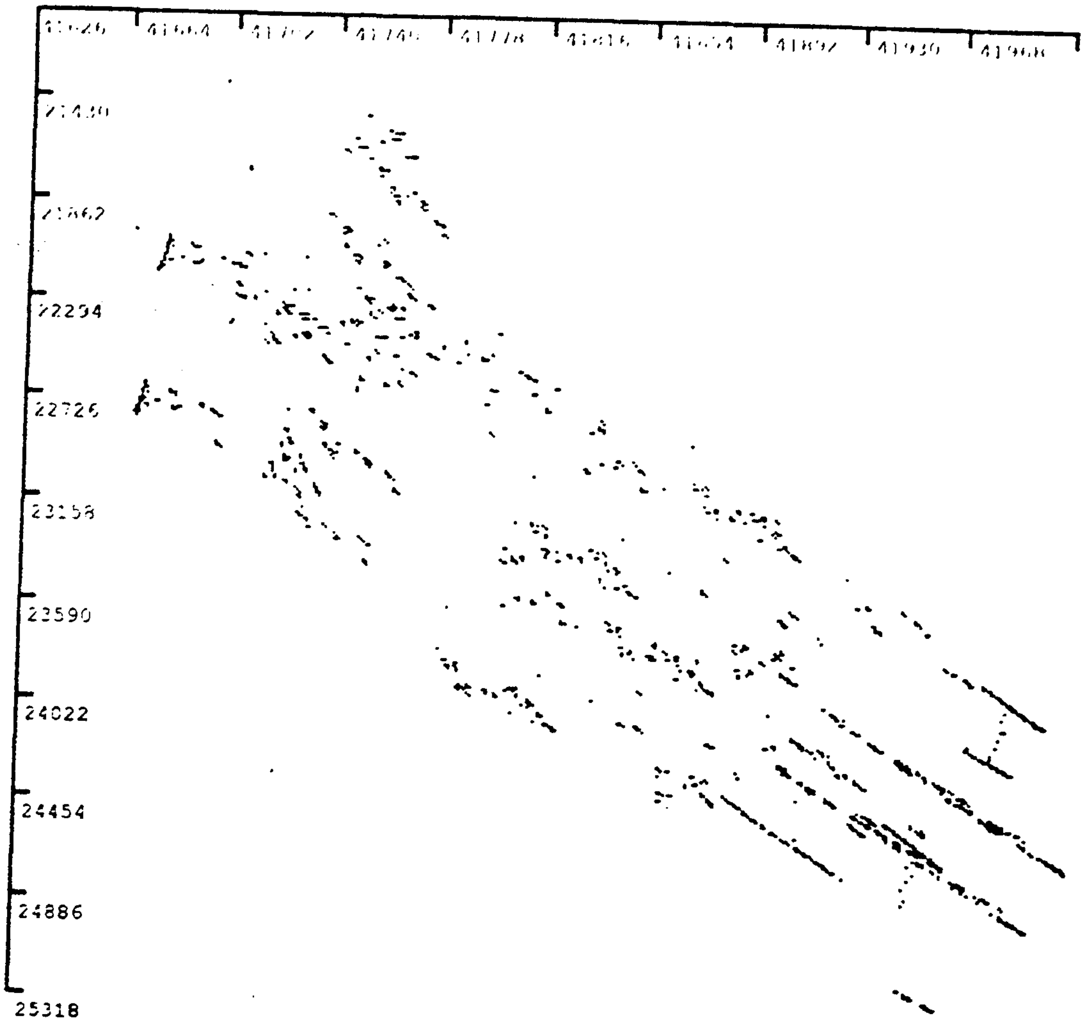

Figure A-70. Input data from data set 22.

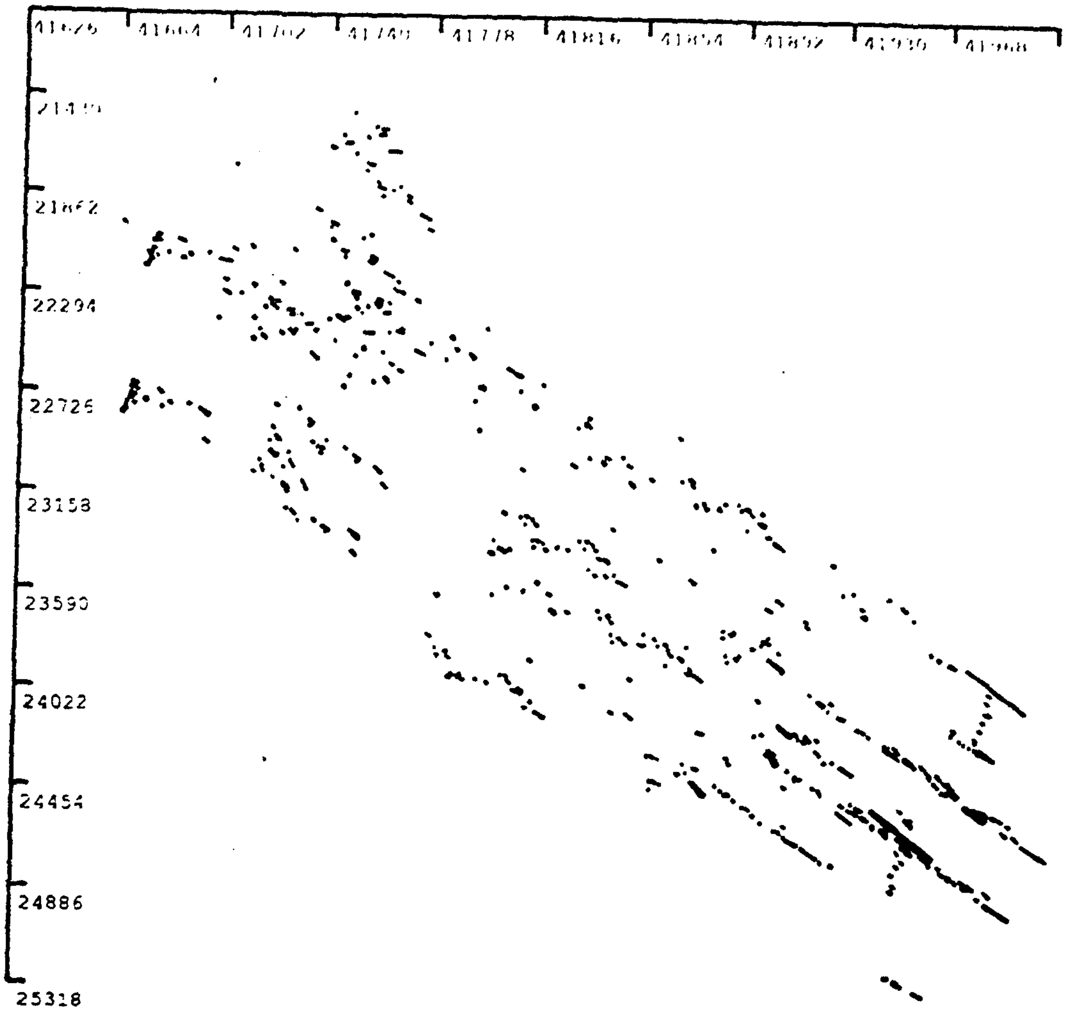

Figure A-71. Initial clusters from data set 22.

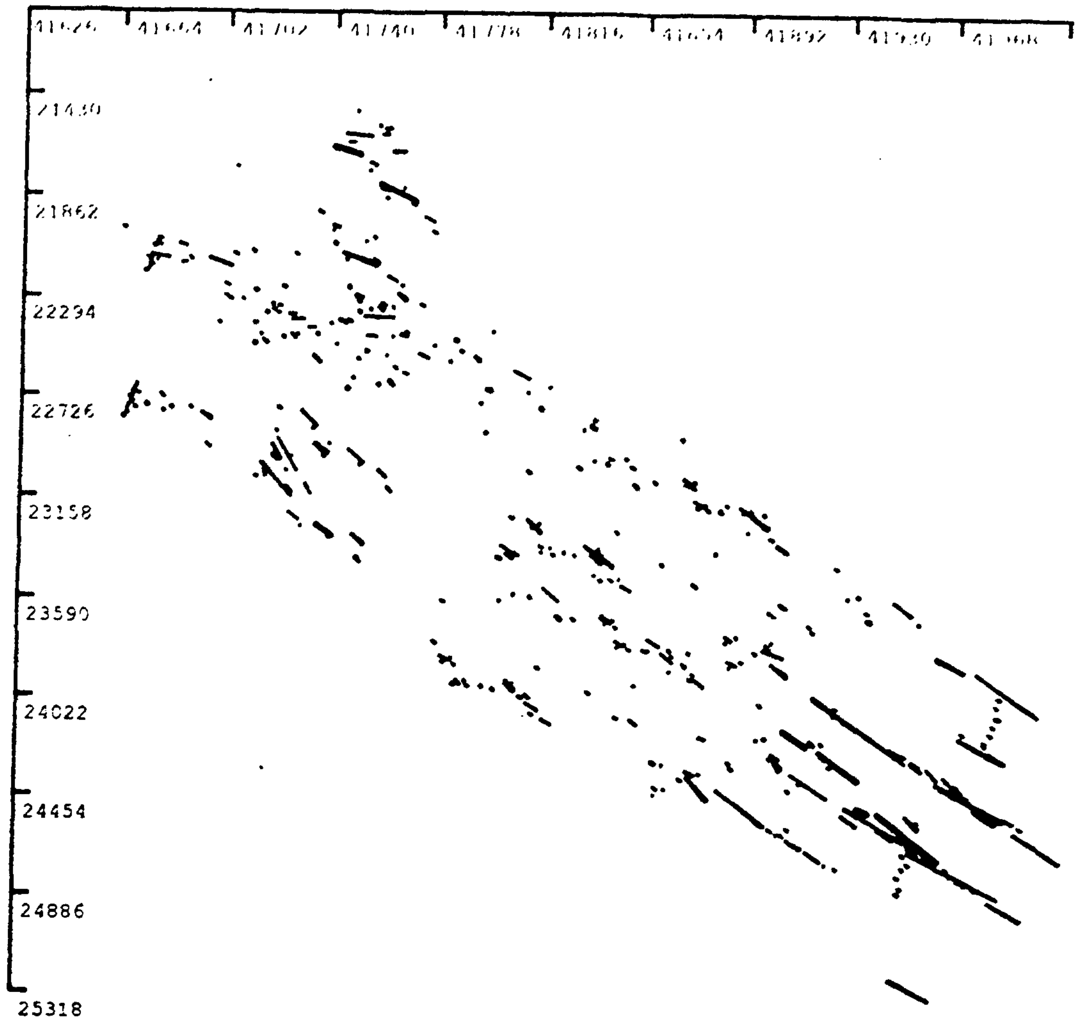

Figure A-72. Linear and online clusters from data set 22.

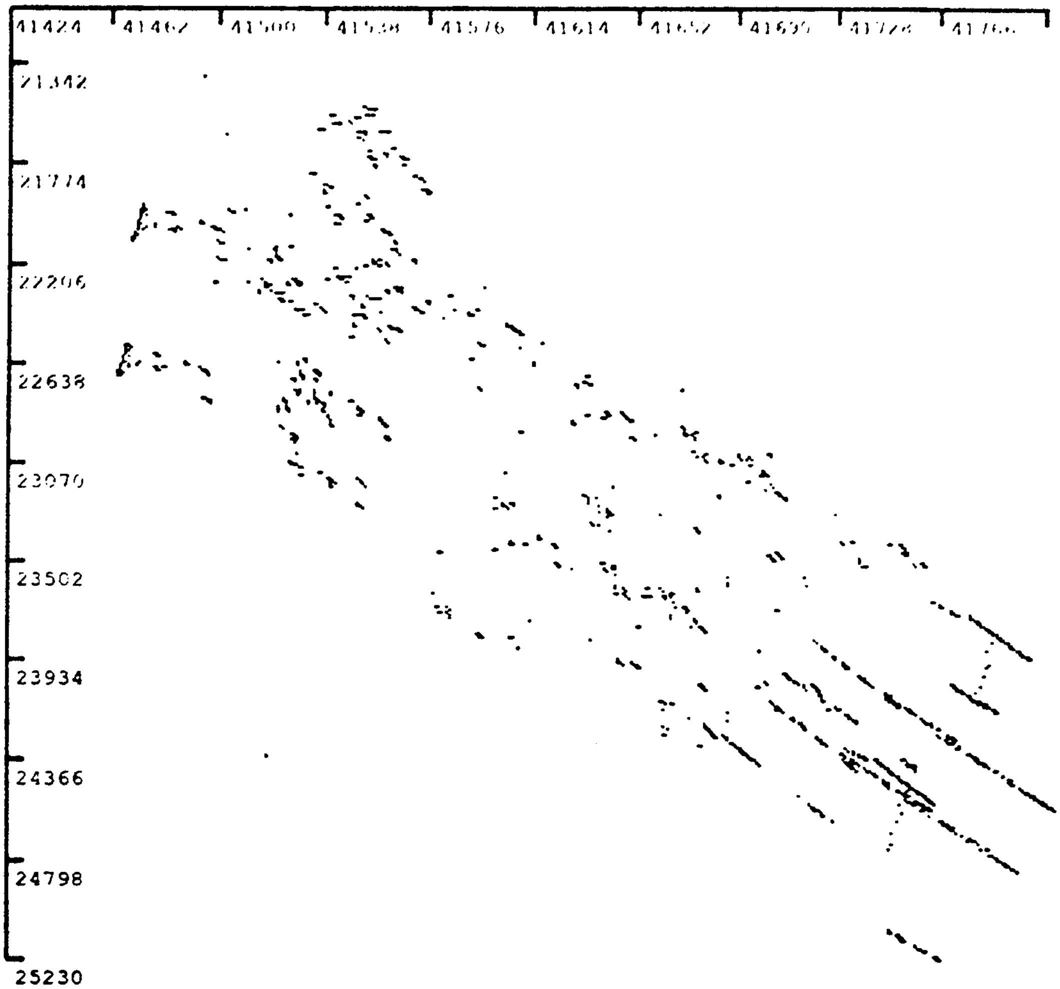

Figure A-73. Input data from data set 23.

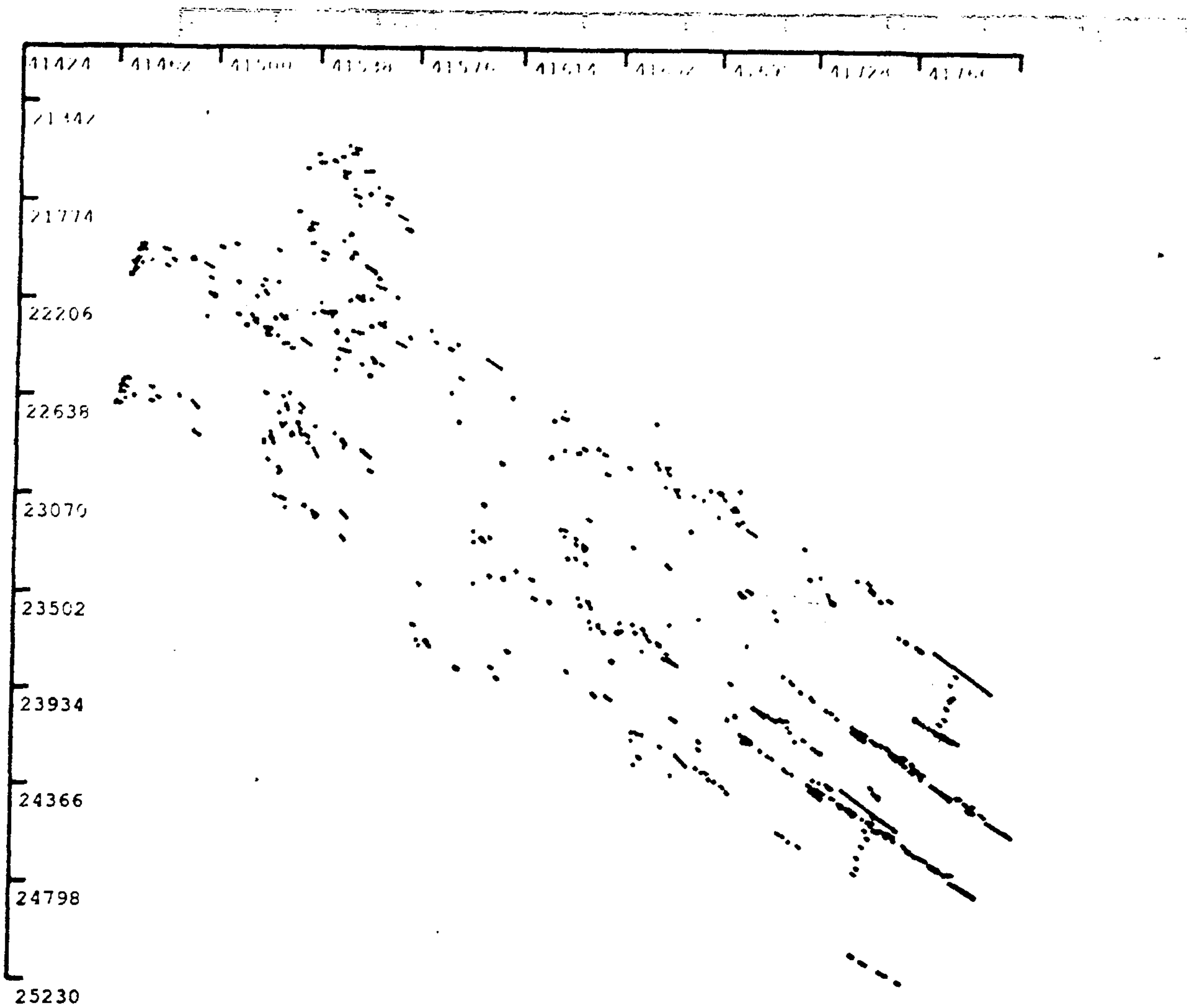

Figure A-74. Initial clusters from data set 23.

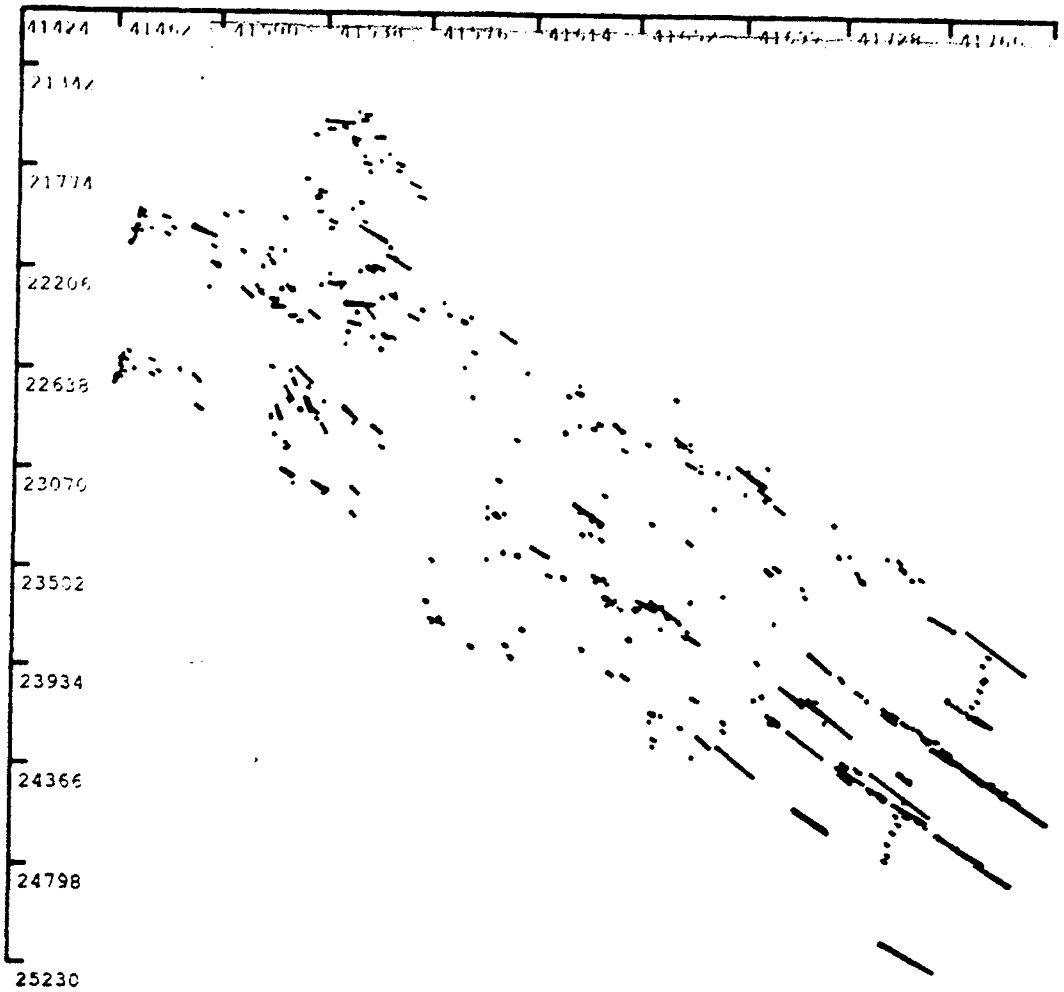

Figure A-75. Linear and online clusters from data set 23.

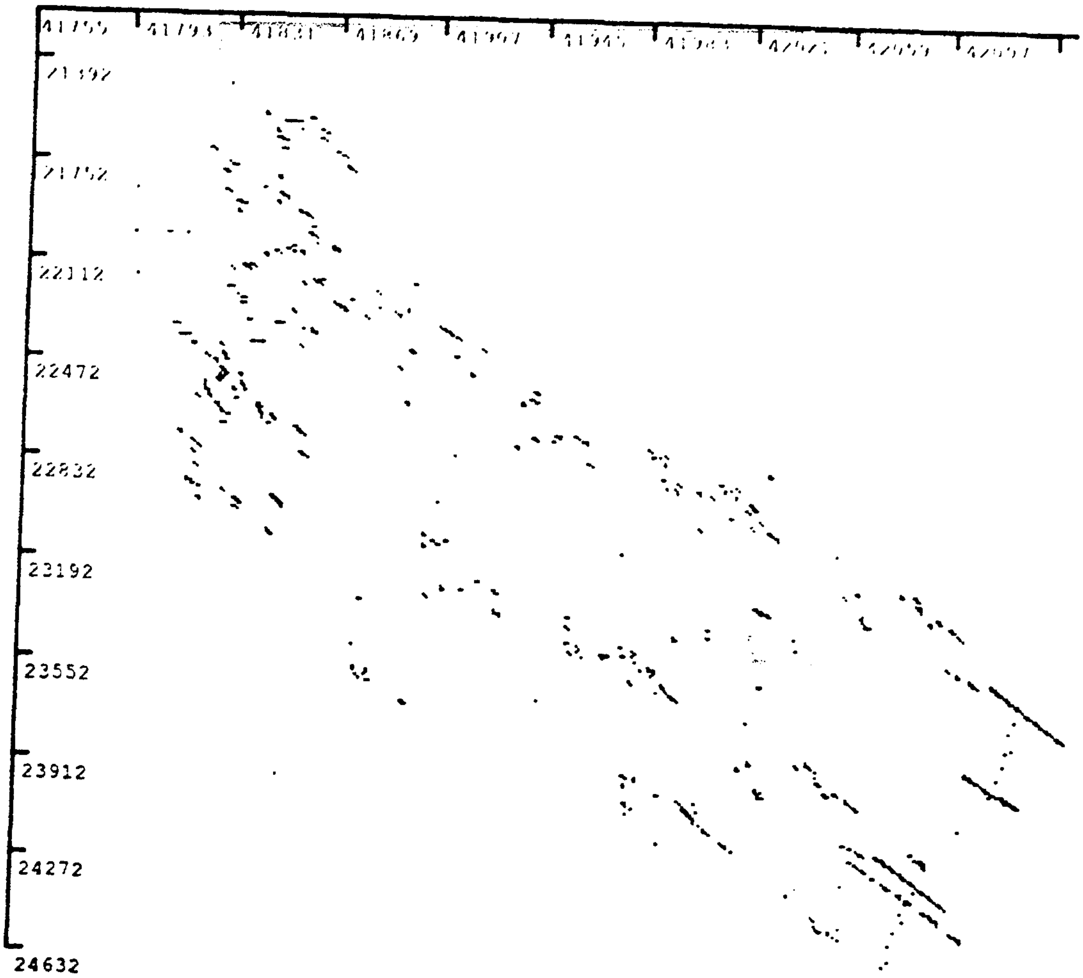

Figure A-76. Input data from data set 24.

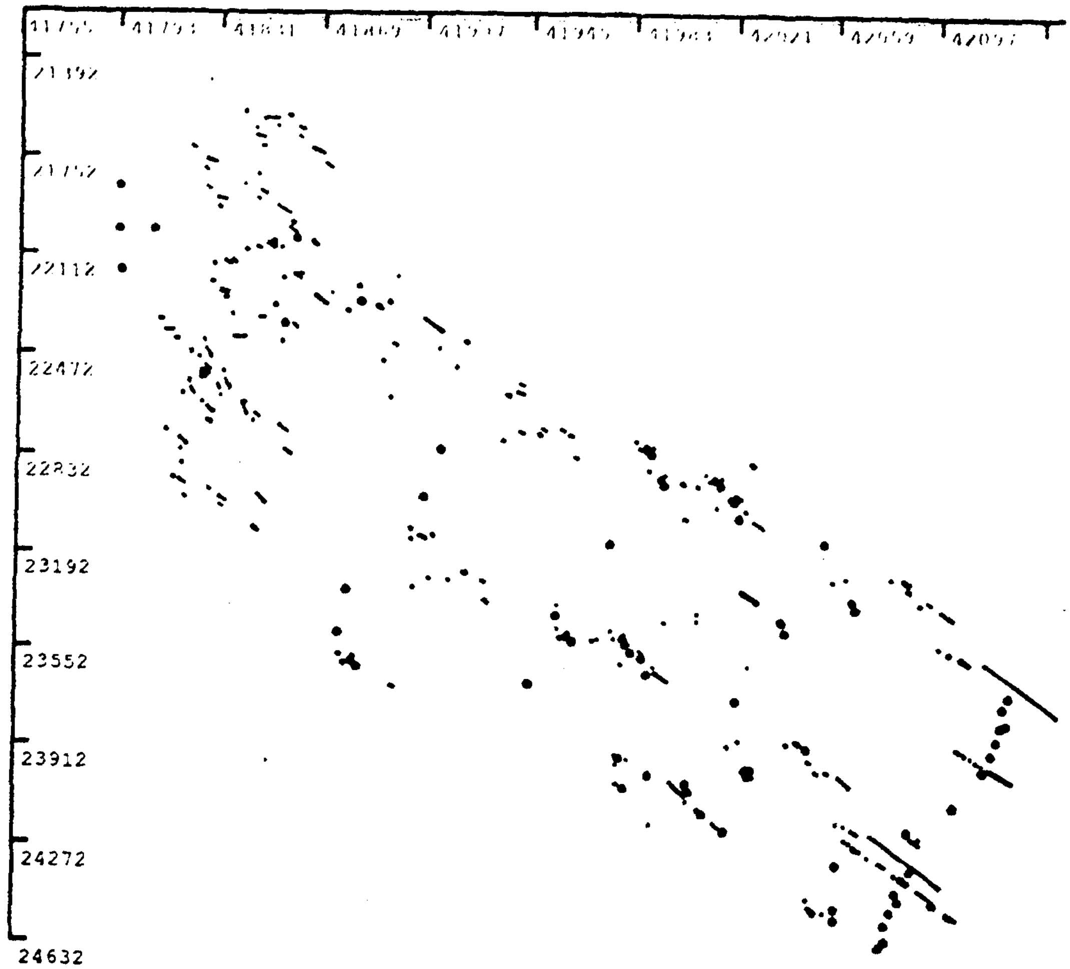

Figure A-77. Initial clusters from data set 24.

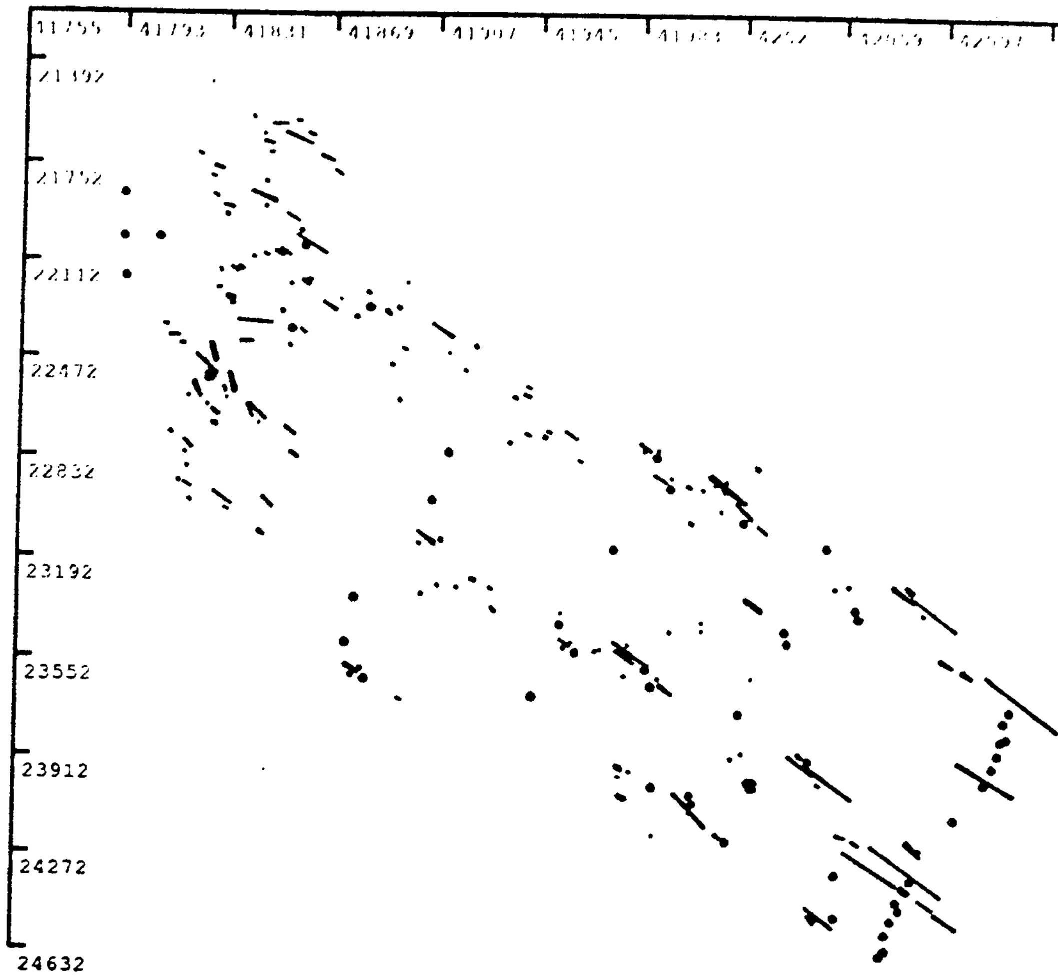

Figure A-78. Linear and online clusters from data set 24.

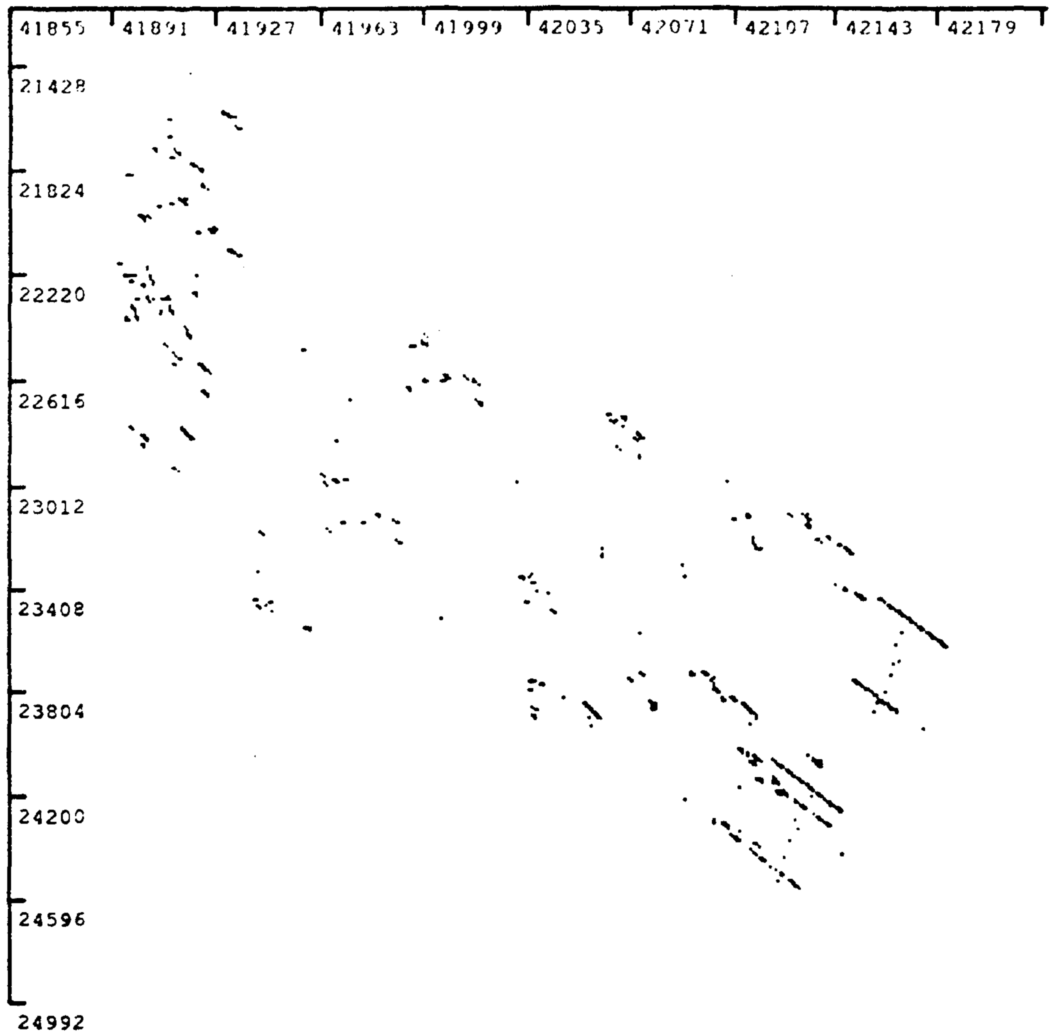

Figure A-79. Input data from data set 25.

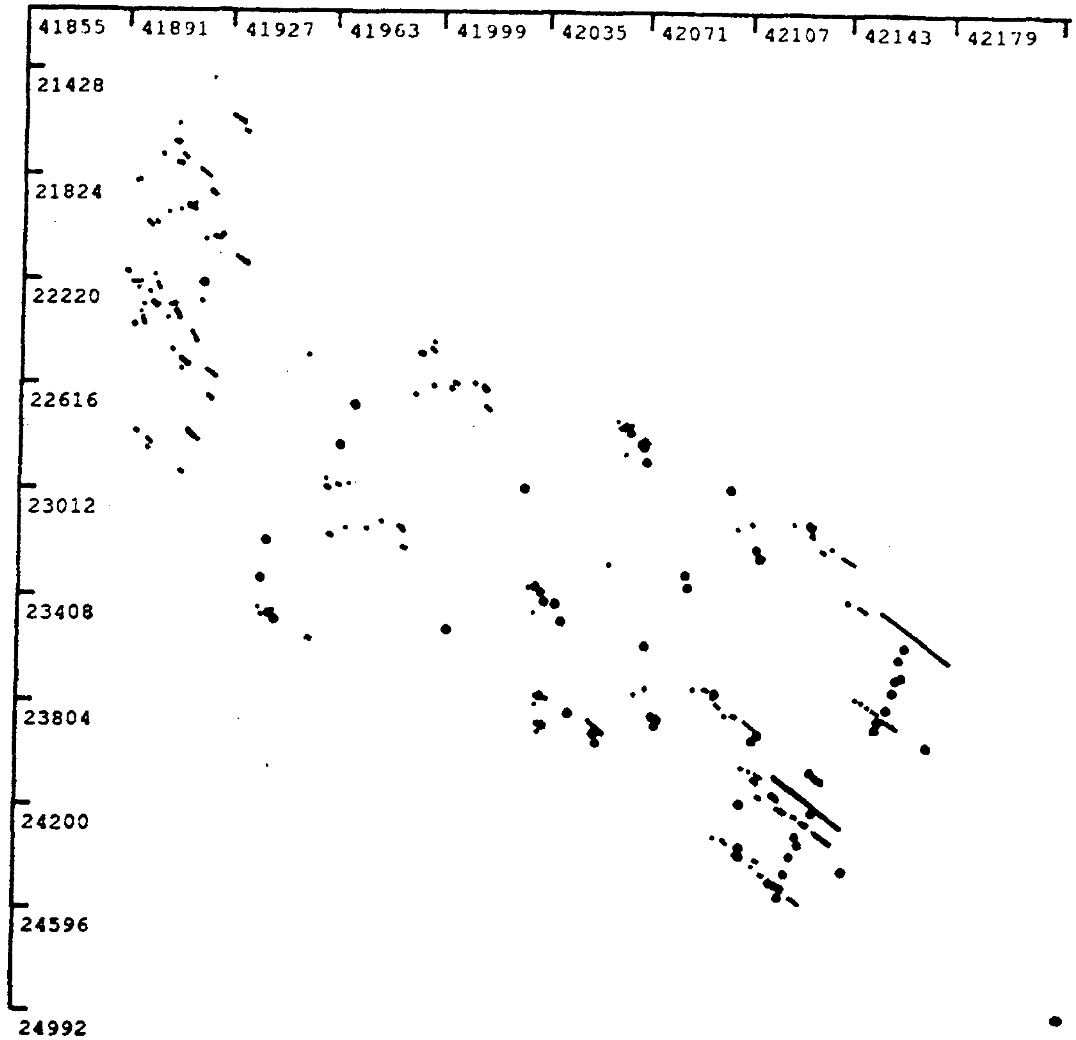

Figure A-80. Initial clusters from data set 25.

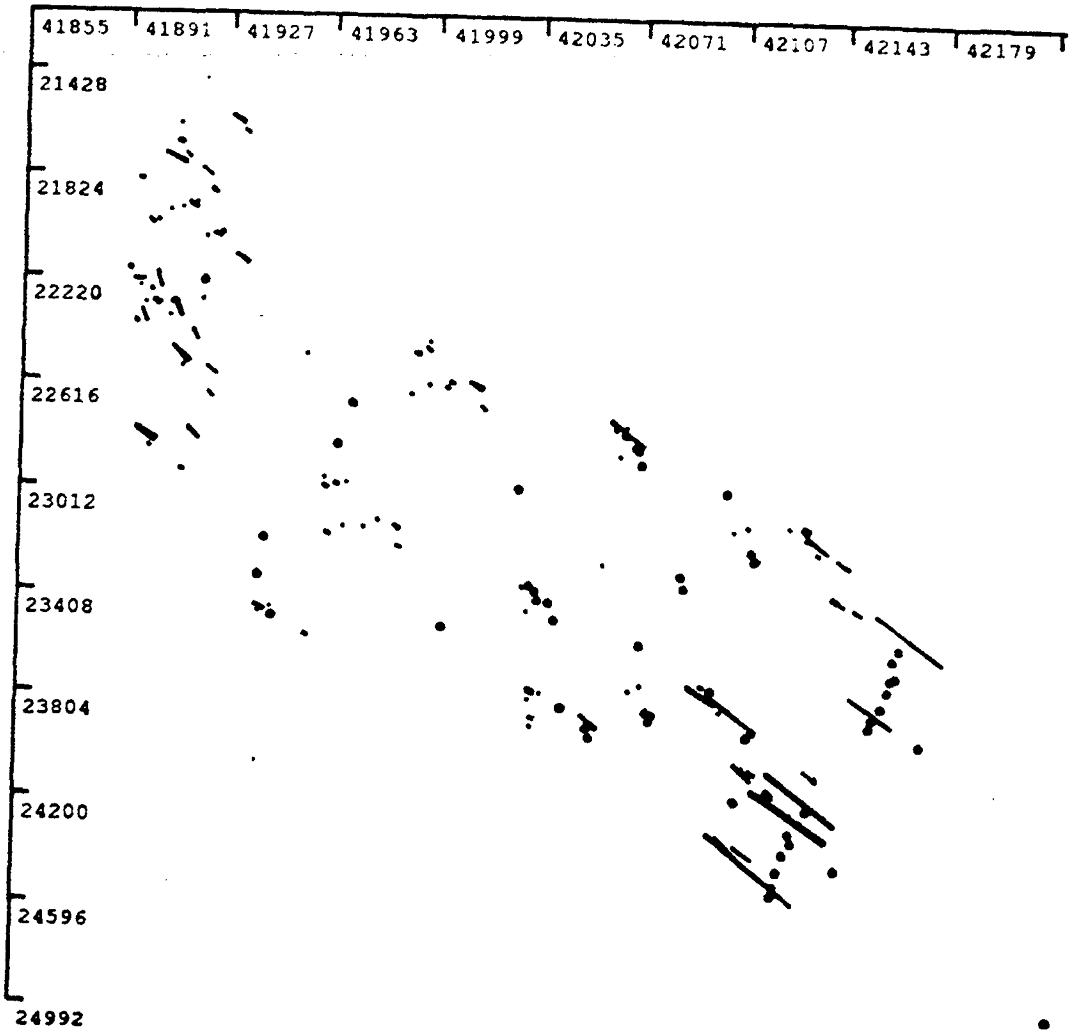

Figure A-81. Linear and online clusters from data set 25.

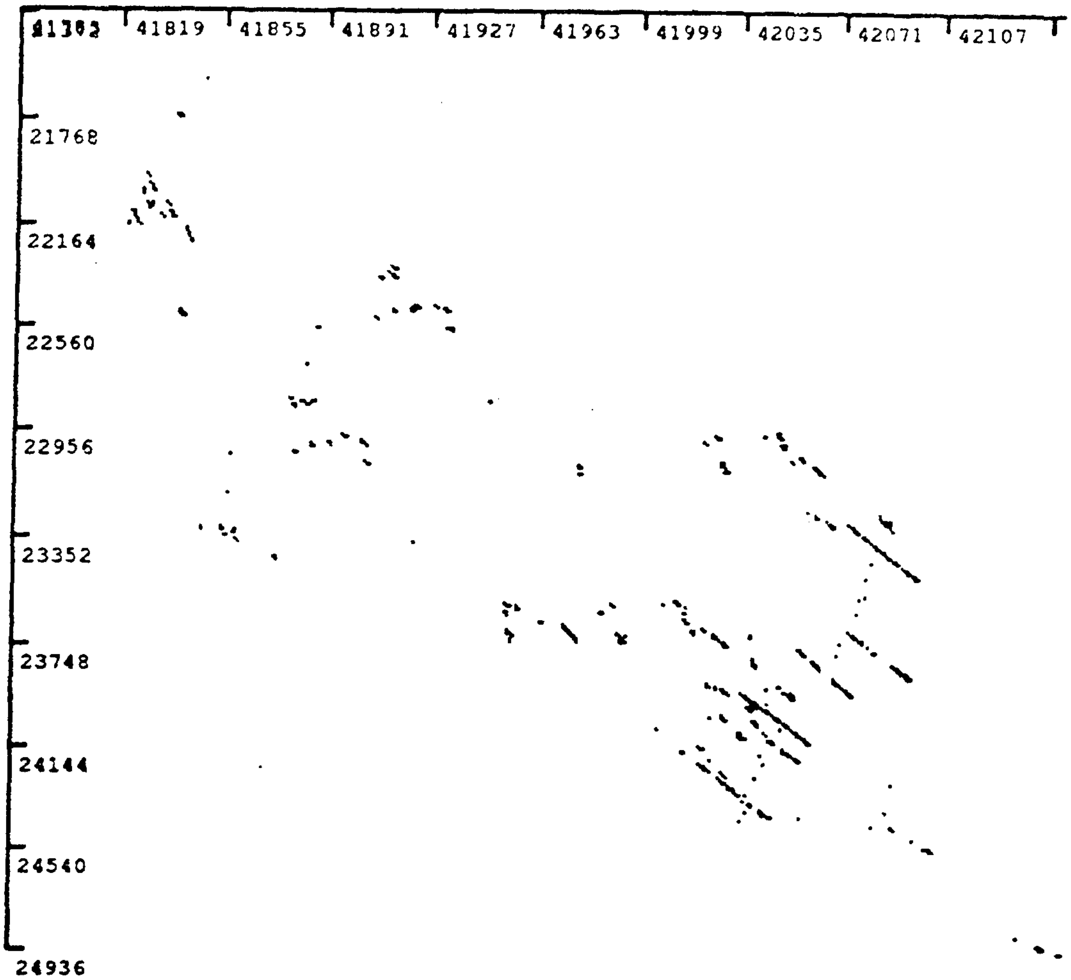

Figure A-82. Input data from data set 26.

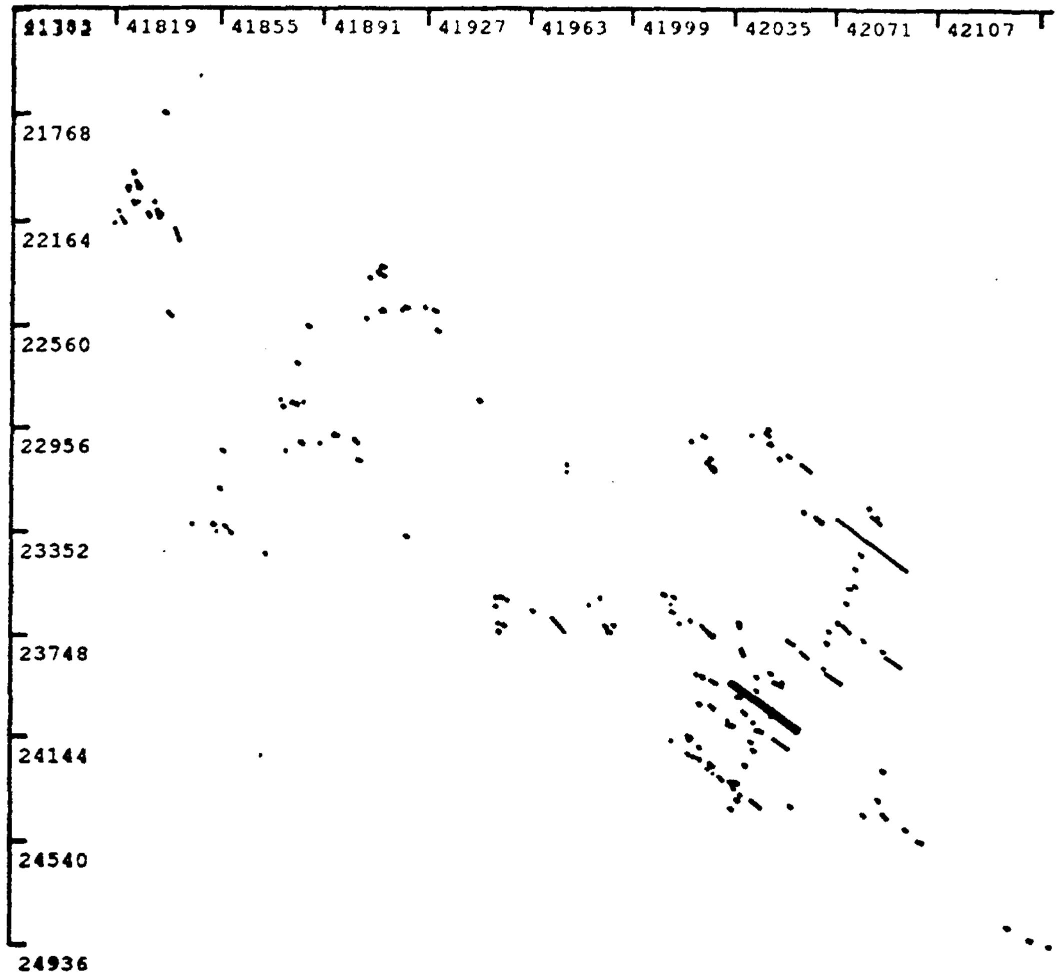

Figure A-83. Initial clusters from data set 26.

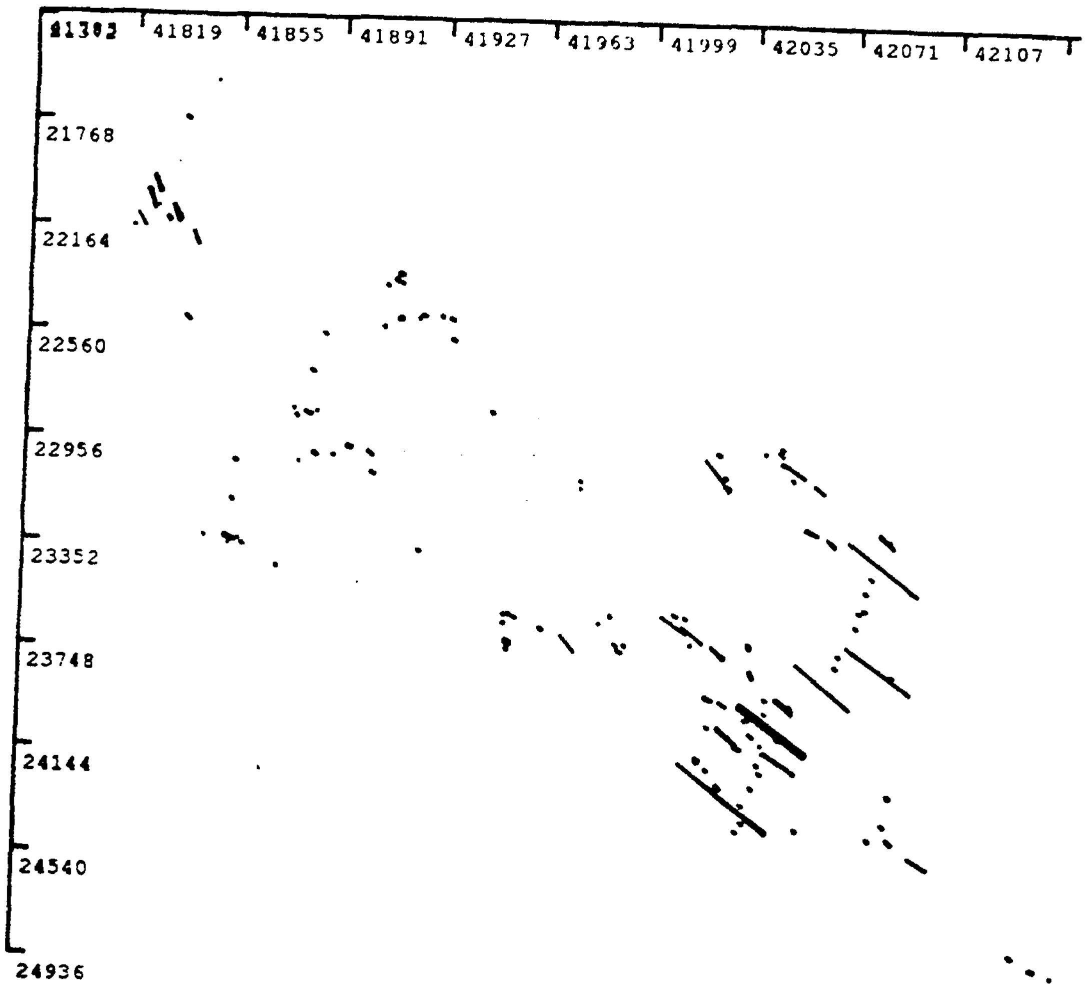

Figure A-84. Linear and online clusters from data set 26.

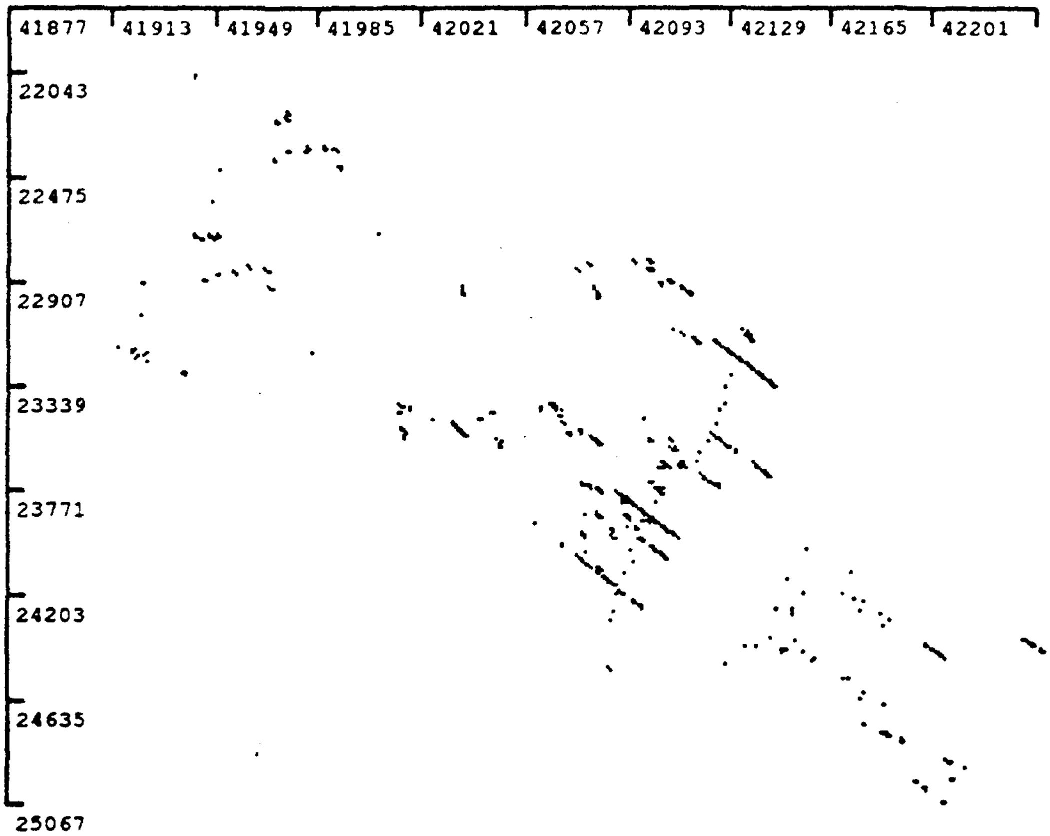

Figure A-85. Input data from data set 27.

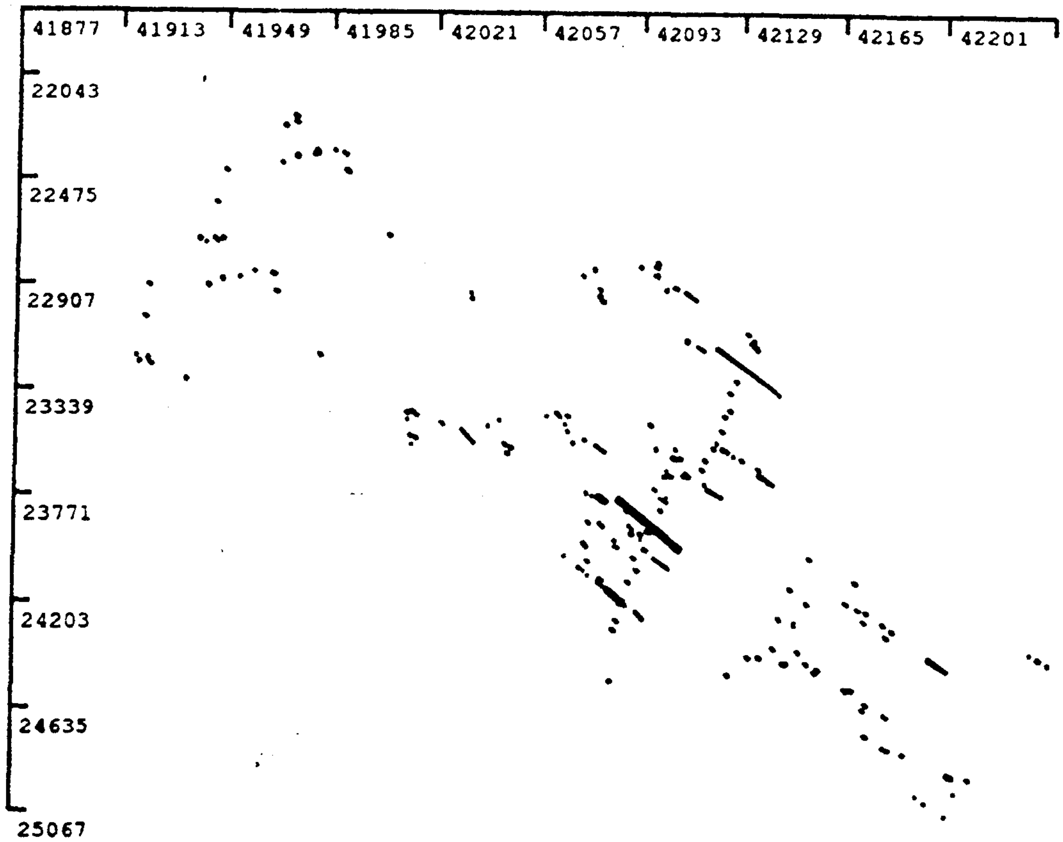

Figure A-86. Initial clusters from data set 27.

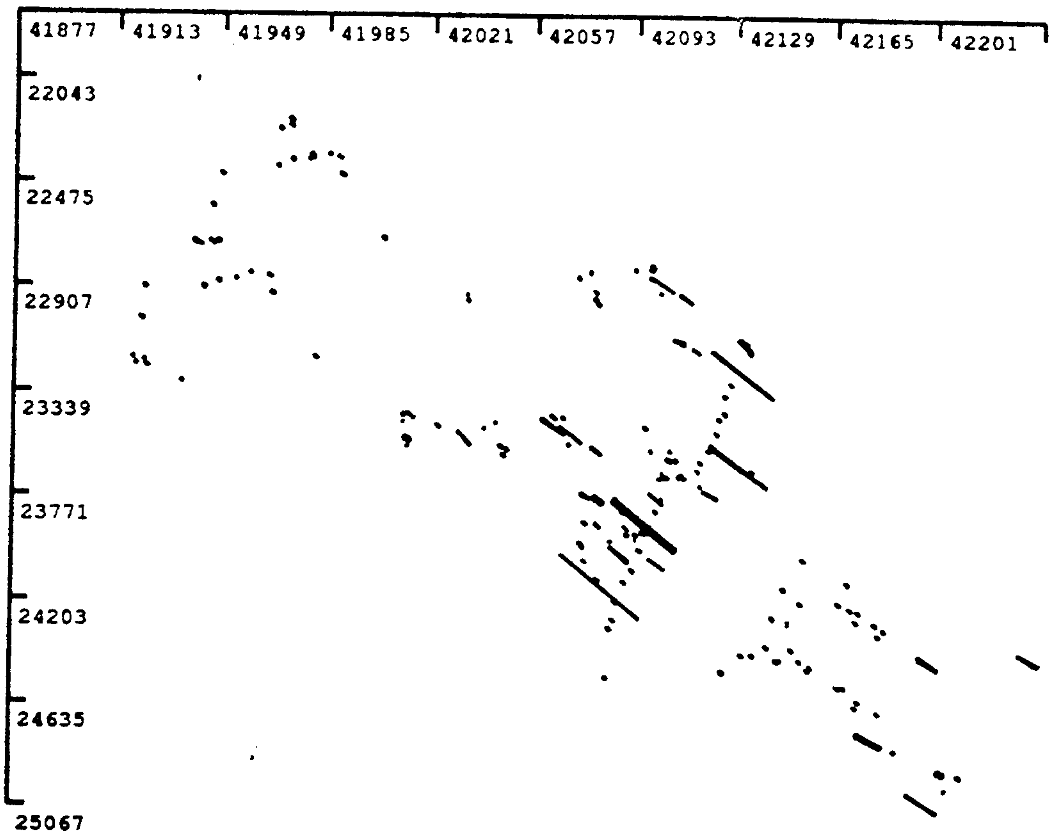

Figure A-87. Linear and online clusters from data set 27.

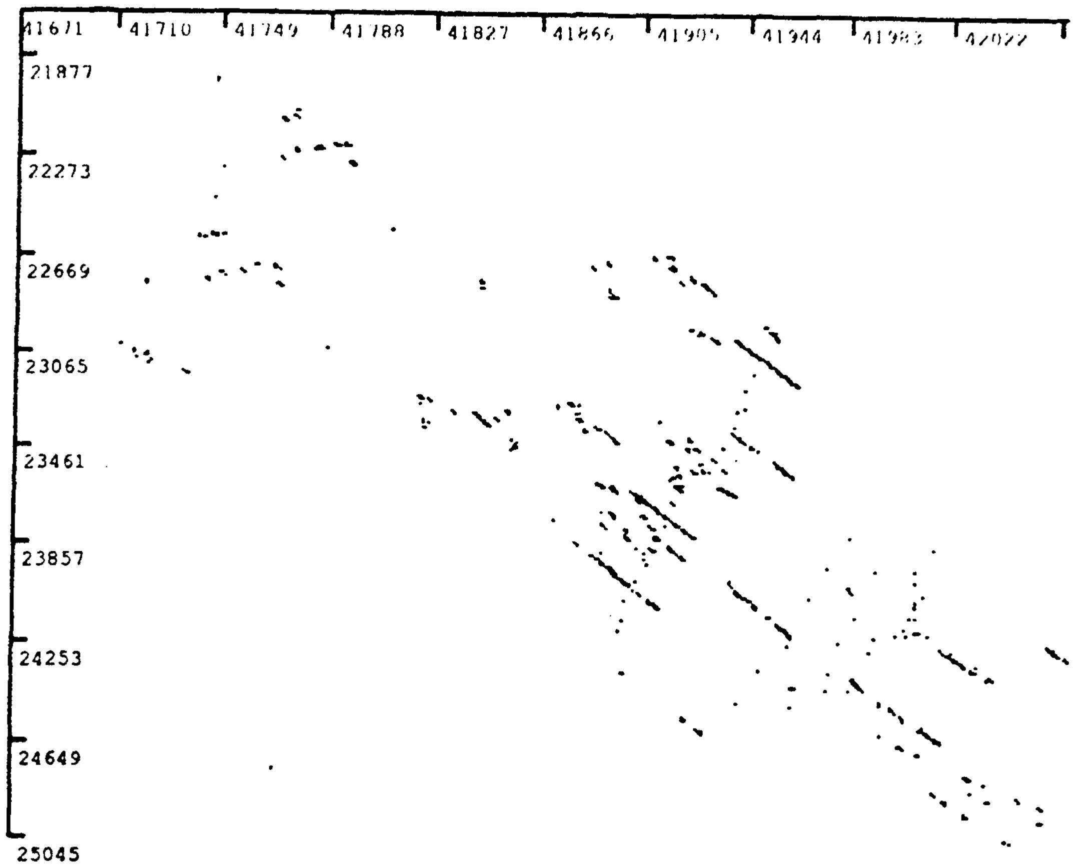

Figure A-88. Input data from data set 28.

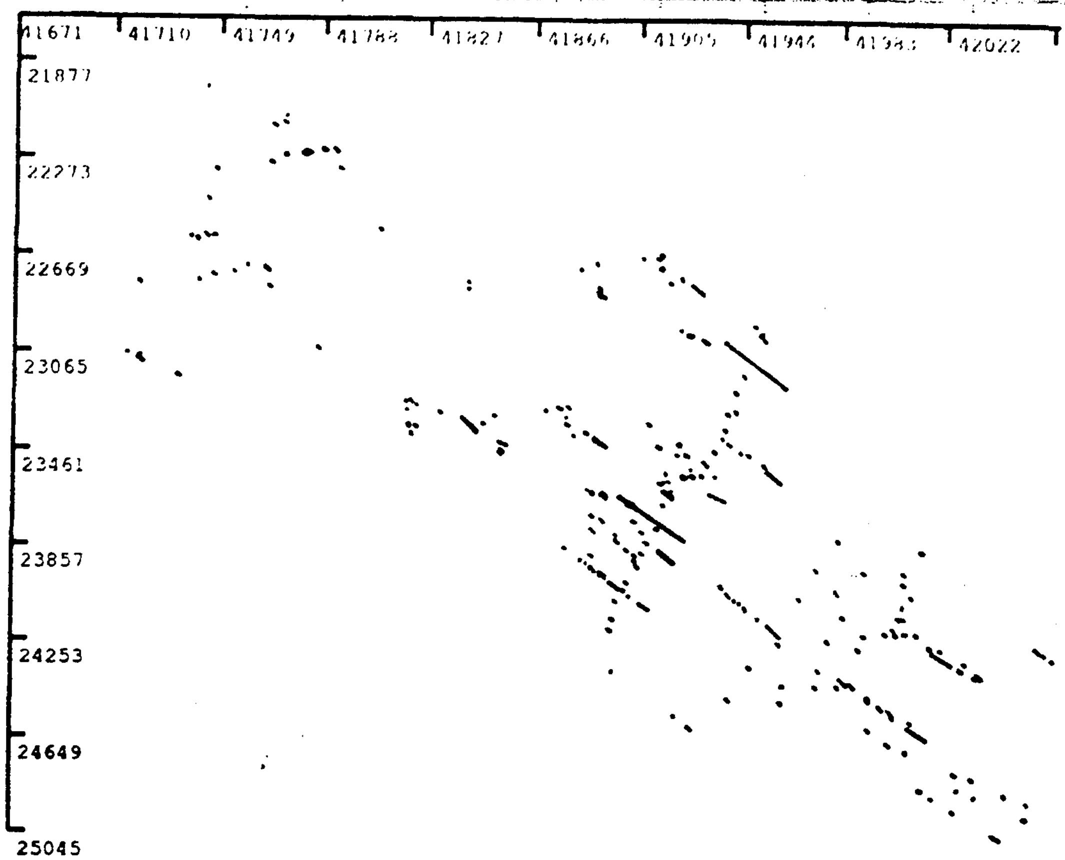

Figure A-89. Initial clusters from data set 28.

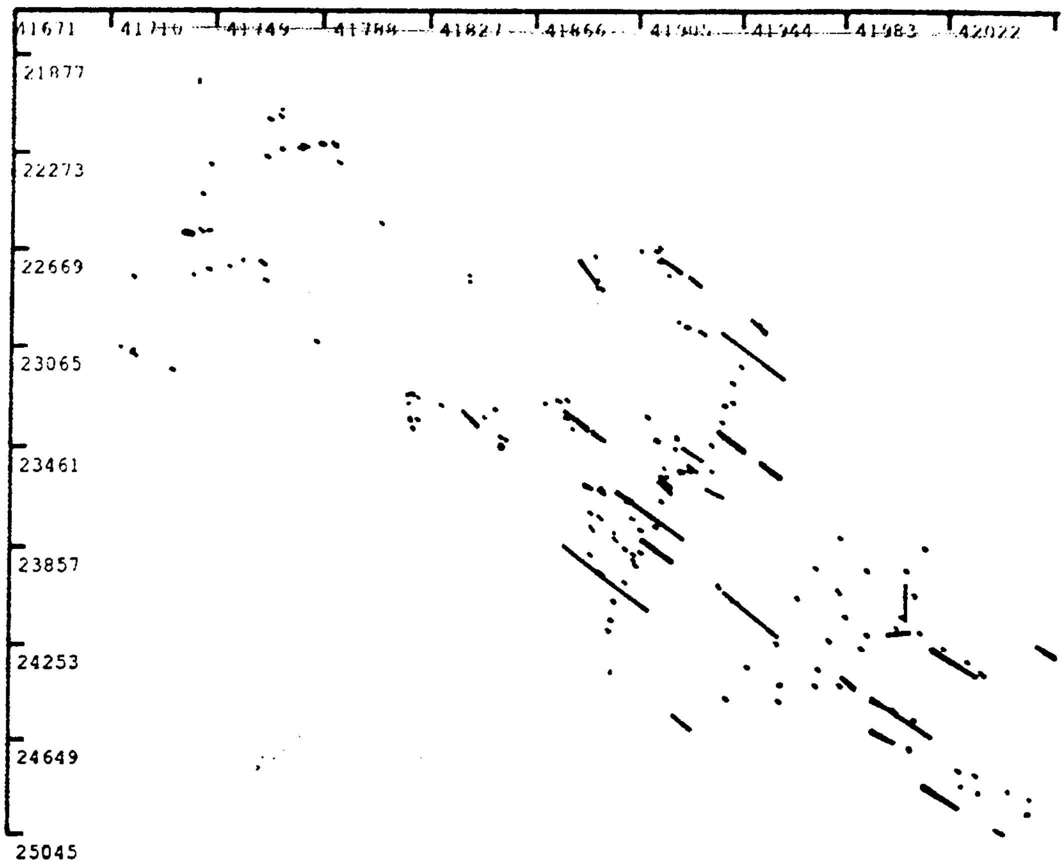

Figure A-90. Linear and online clusters from data set 28.

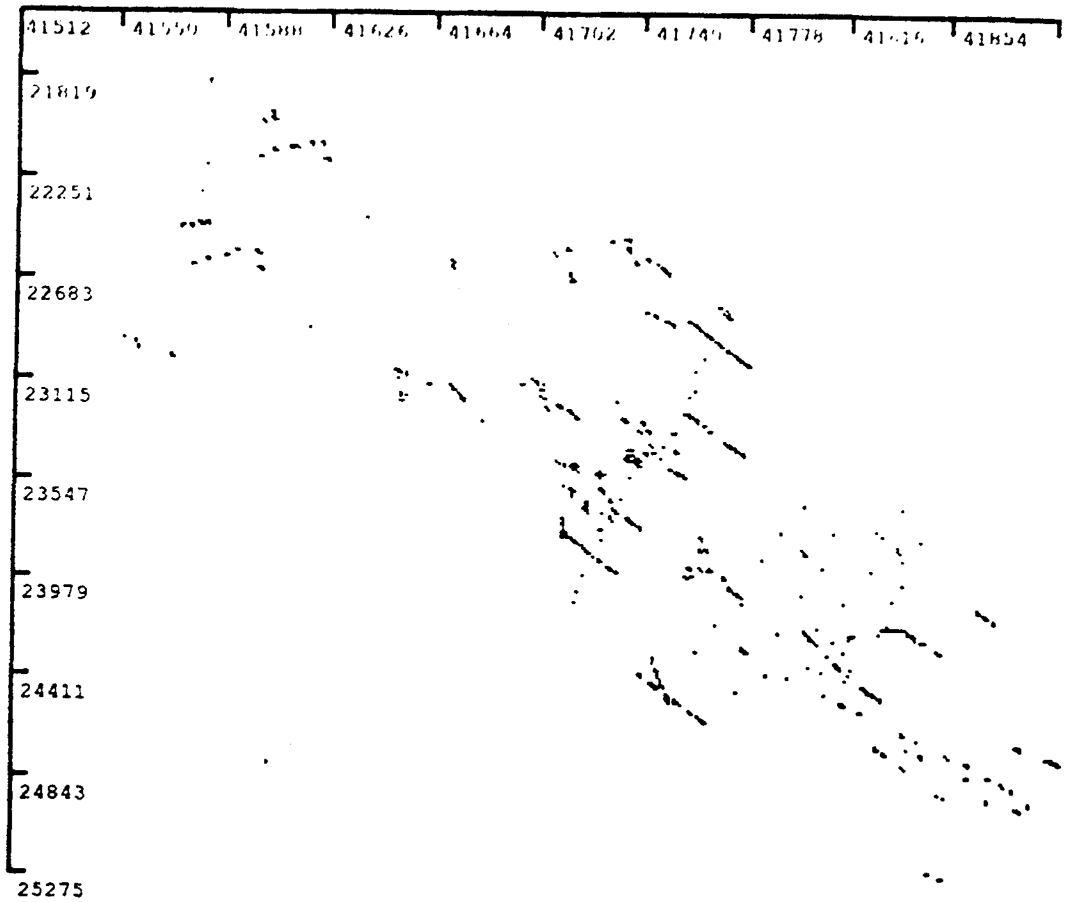

Figure A-91. Input data from data set 29.

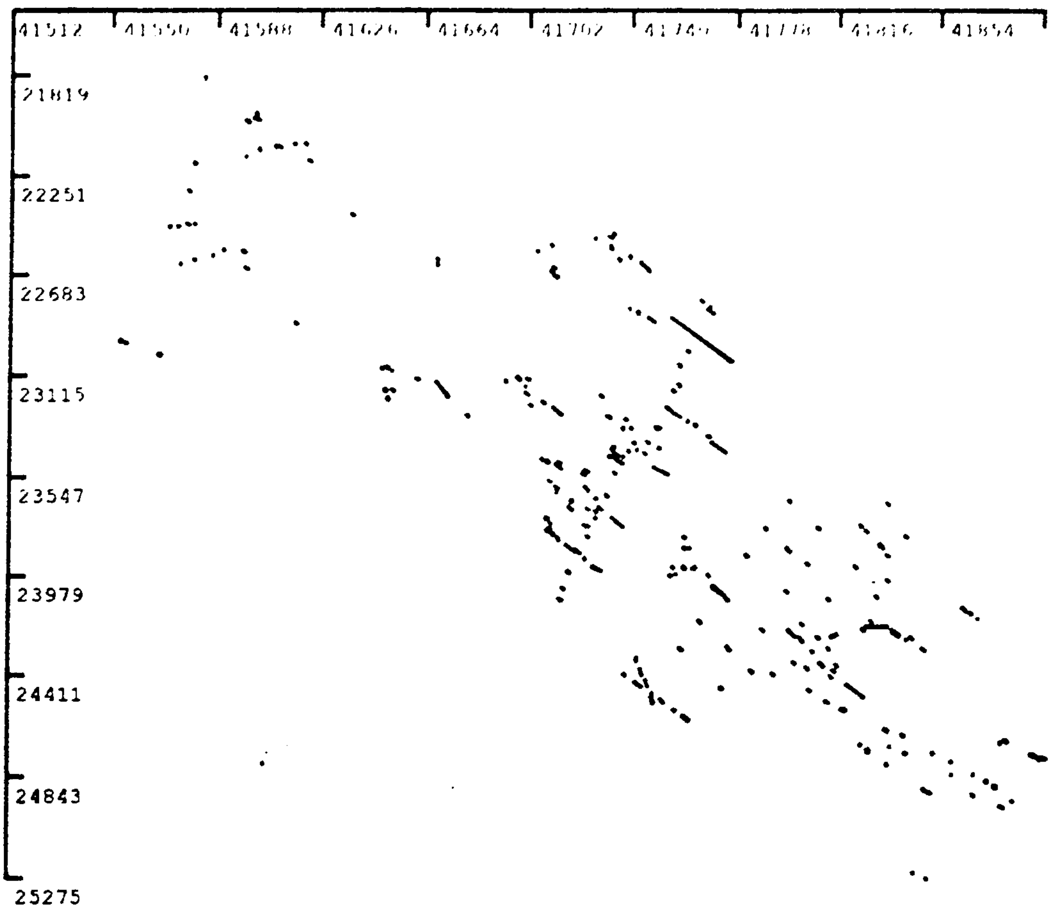

Figure A-92. Initial clusters from data set 29.

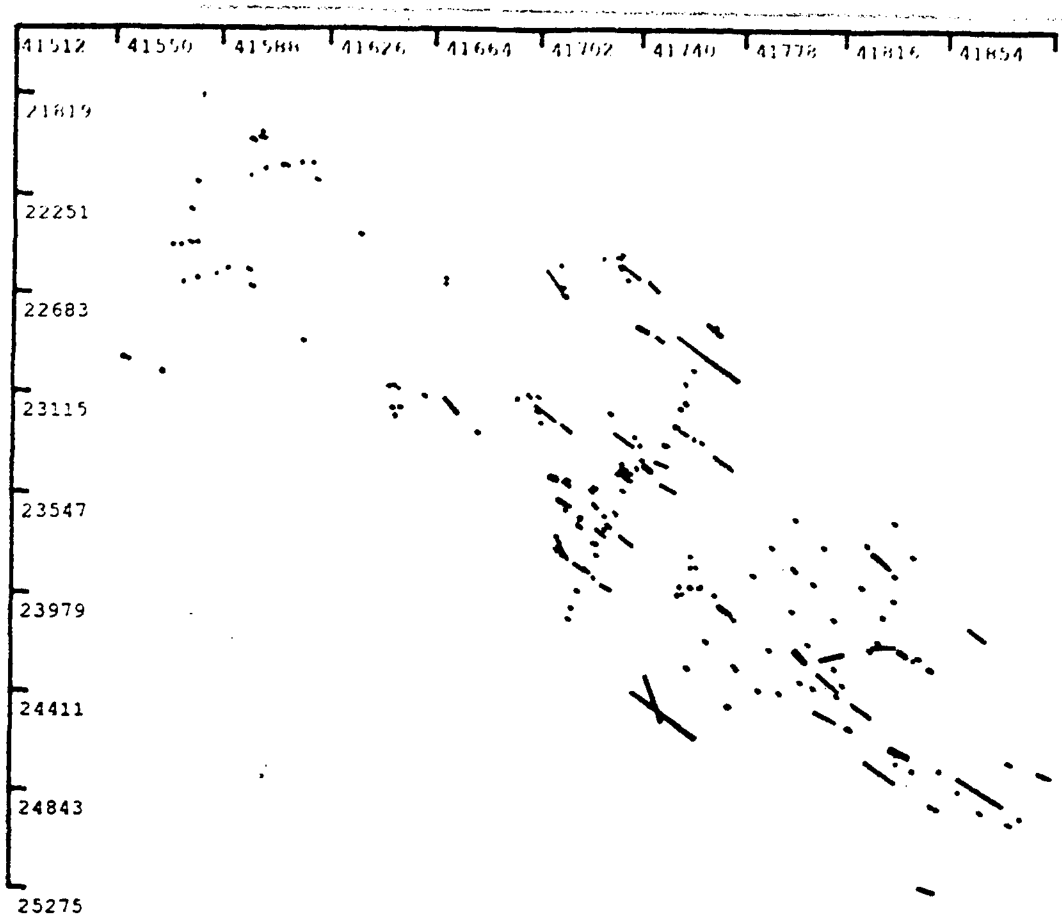

Figure A-93. Linear and online clusters from data set 29.

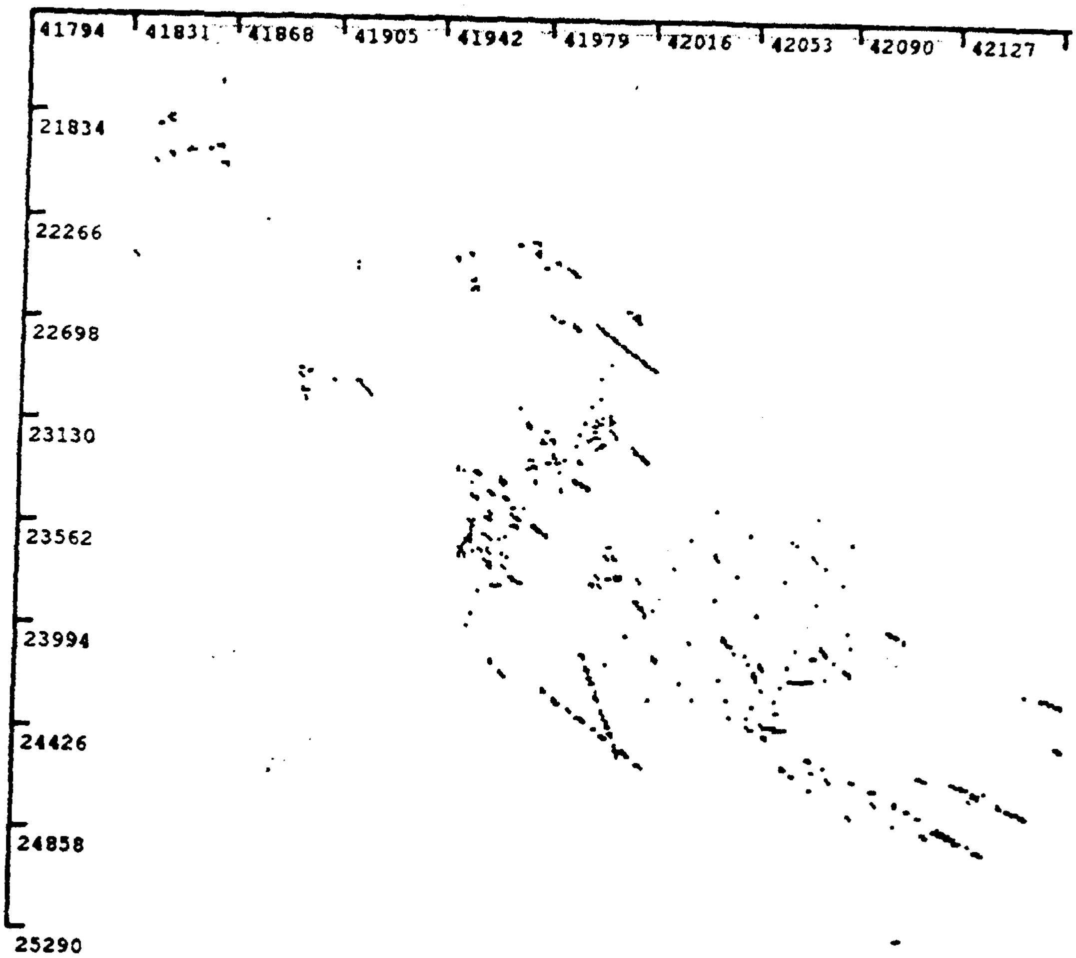

Figure A-94. Input data from data set 30.

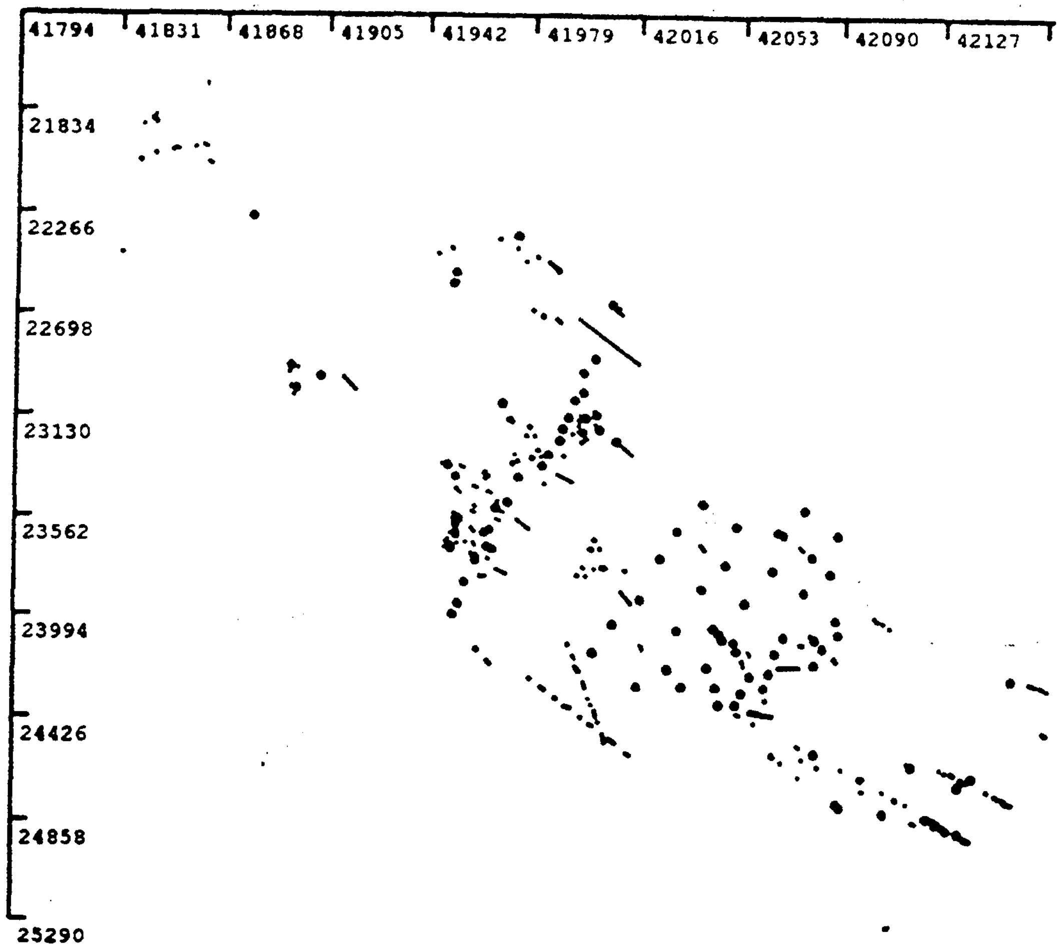

Figure A-95. Initial clusters from data set 30.

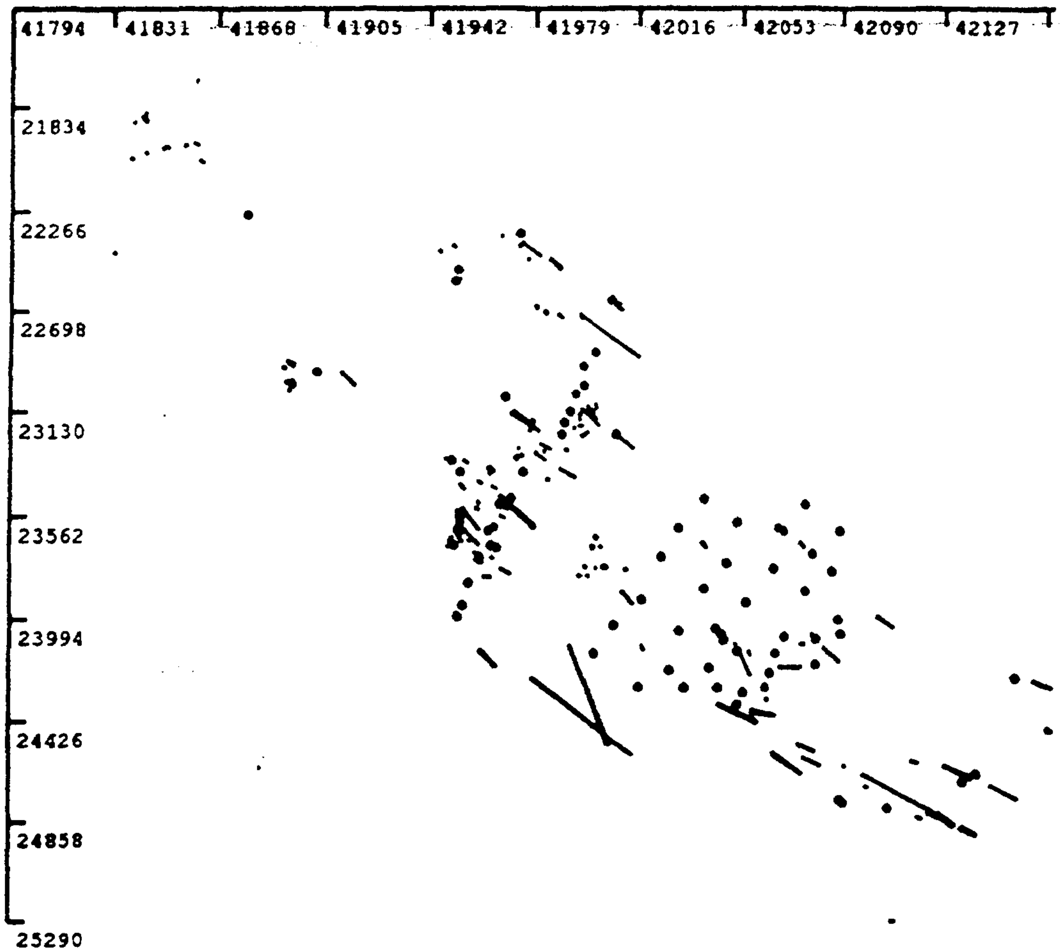

Figure A-96. Linear and online clusters from data set 30.

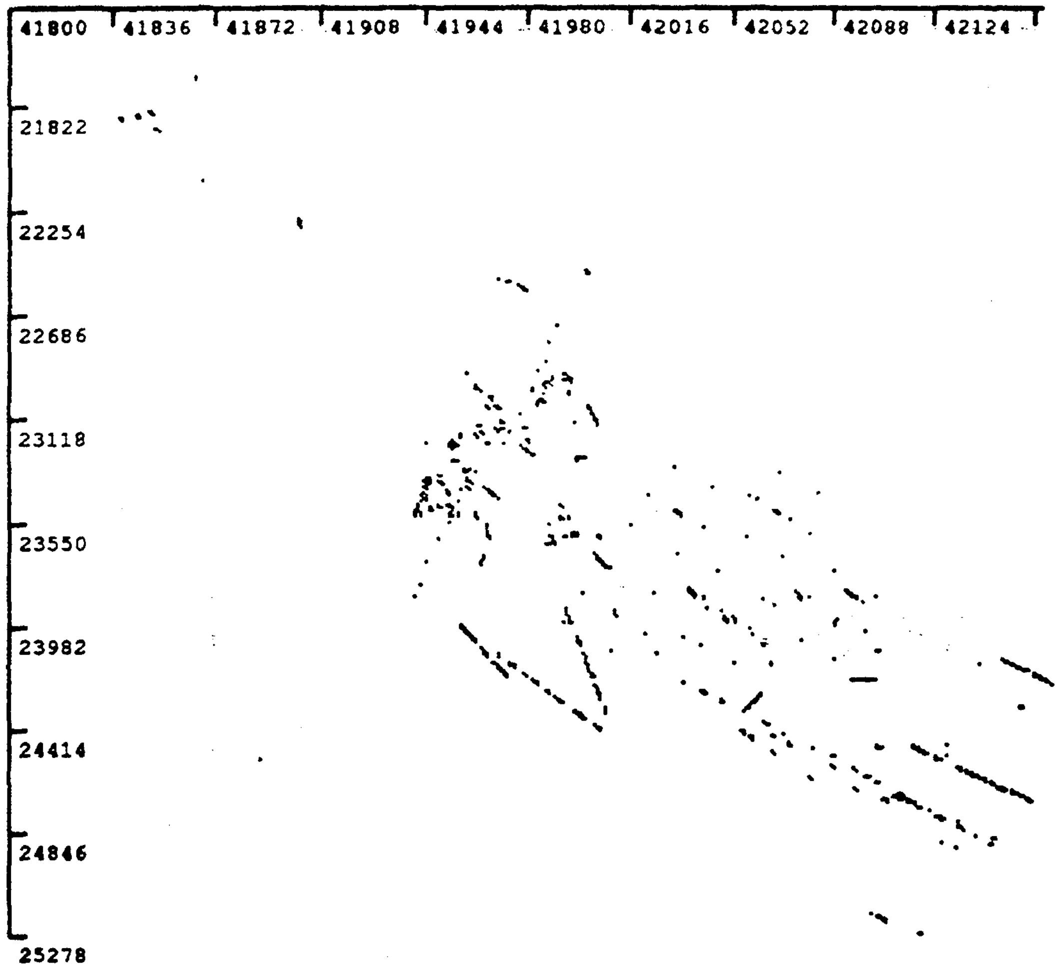

Figure A-97. Input data from data set 31.

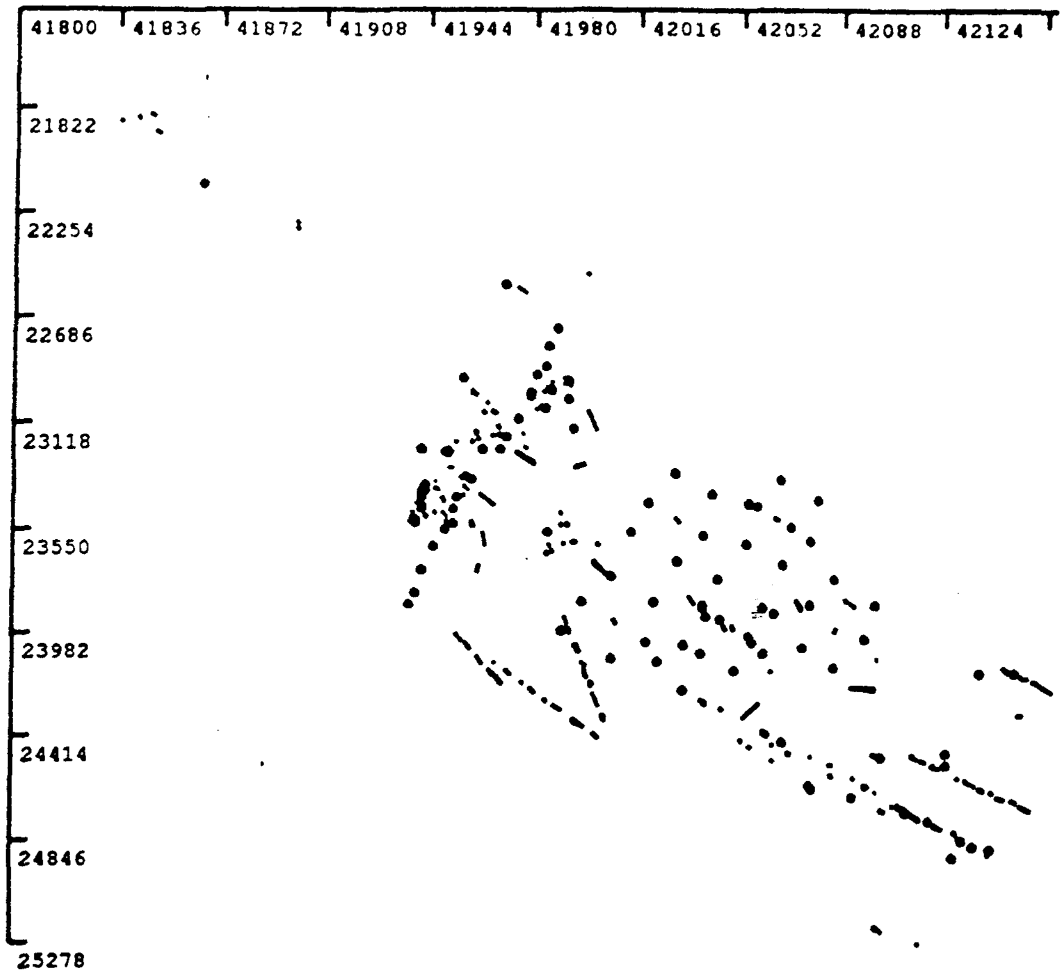

Figure A-98. Initial clusters from data set 31.

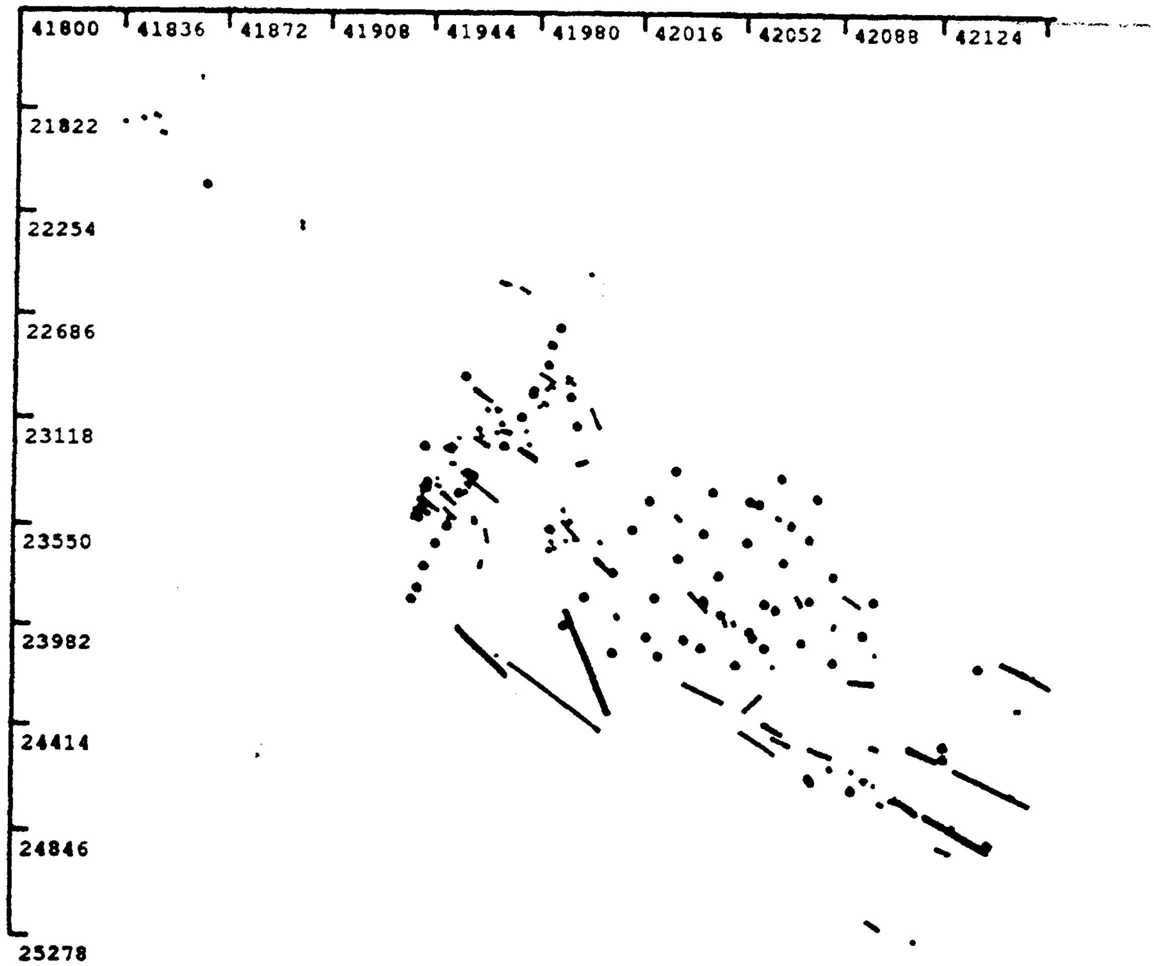

Figure A-99. Linear and online clusters from data set 31.

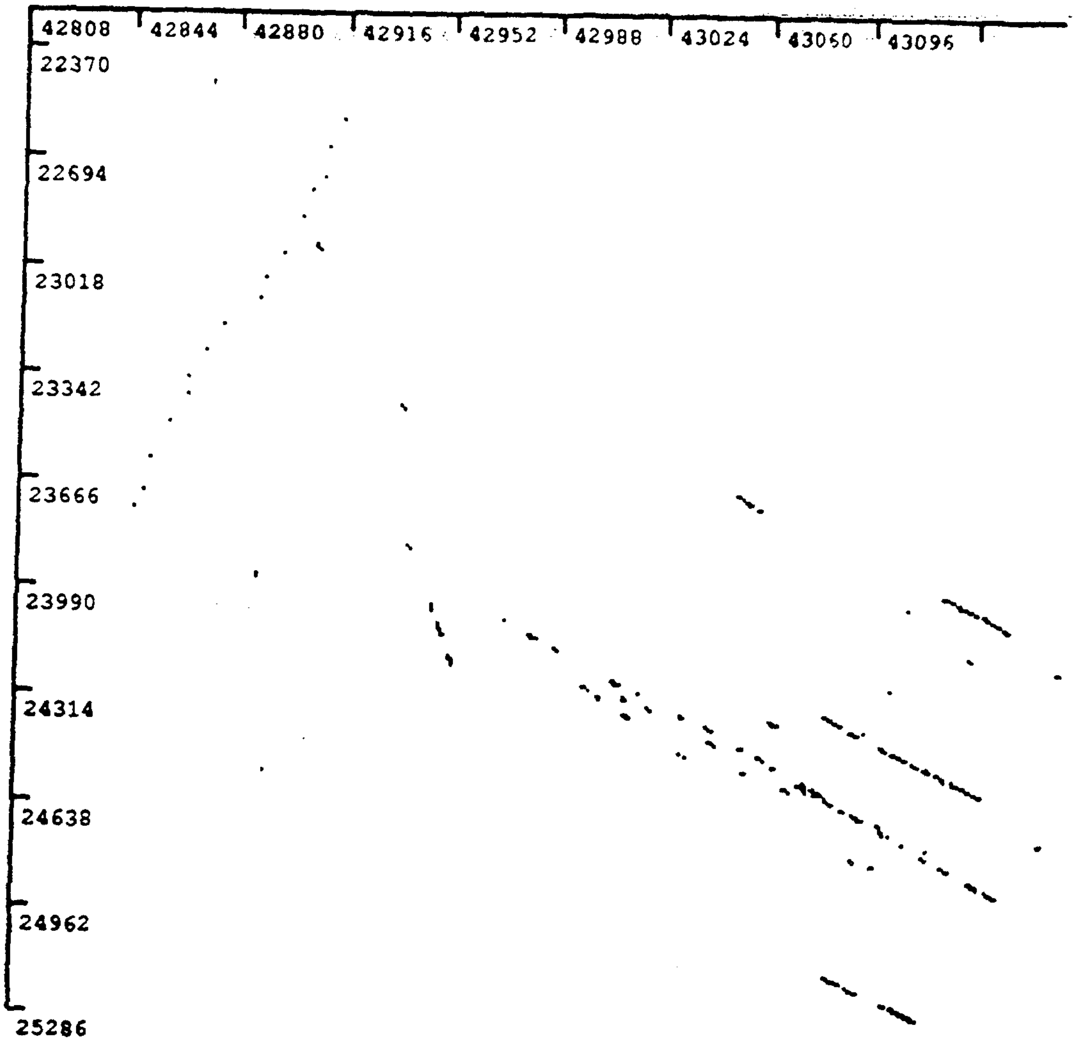

Figure A-100. Input data from data set 32.

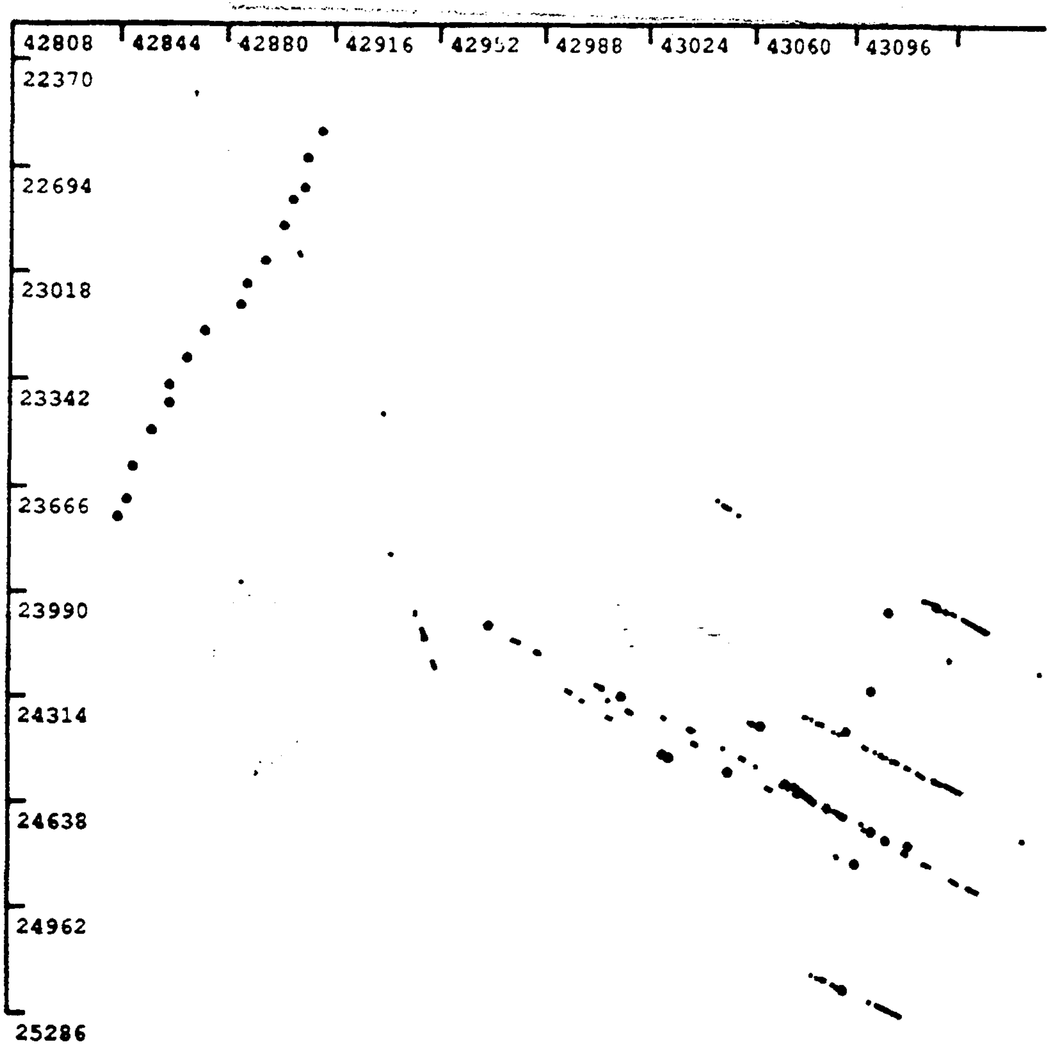

Figure A-101. Initial clusters from data set 32.

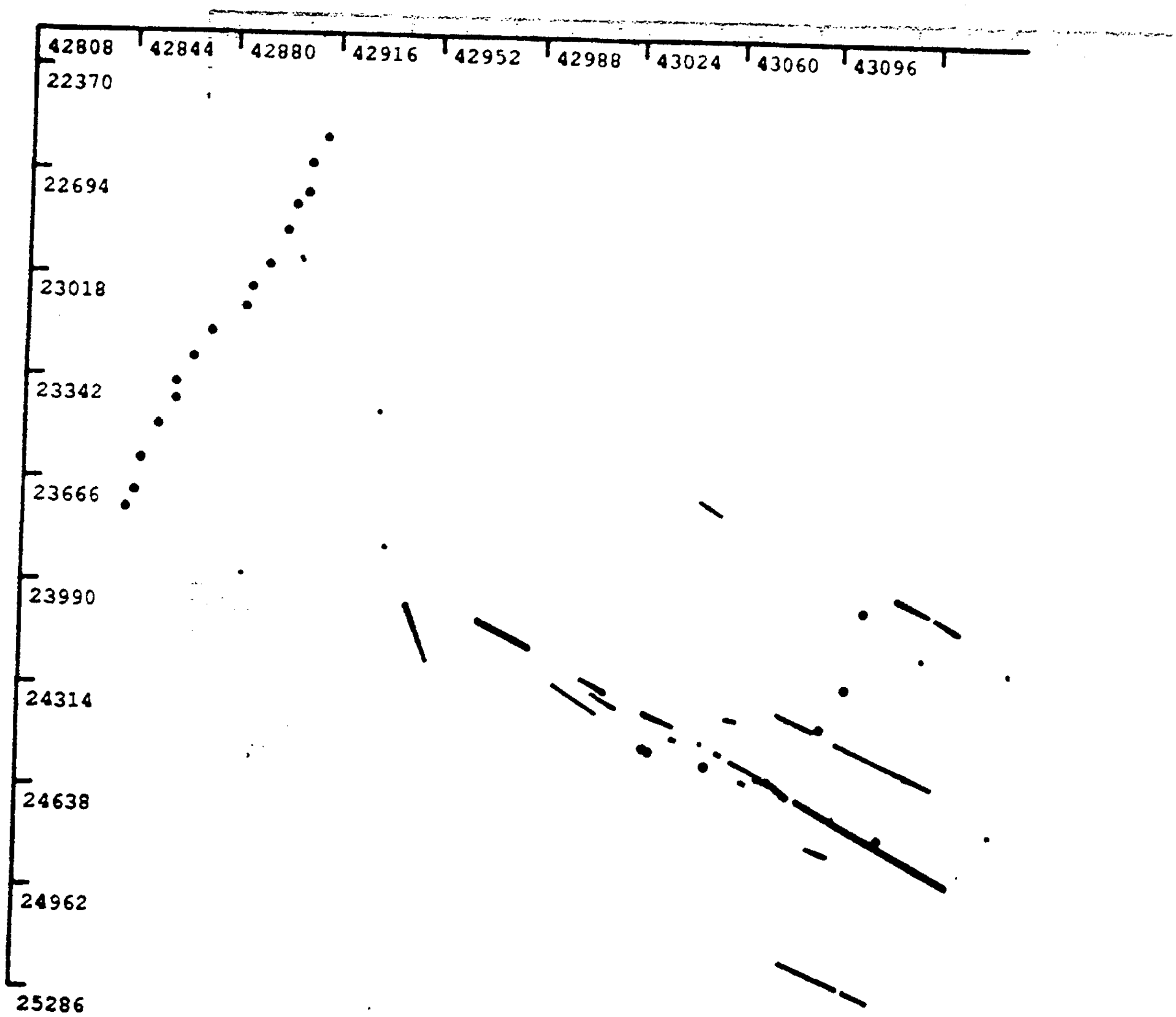

Figure A-102. Linear and online clusters from data set 32.

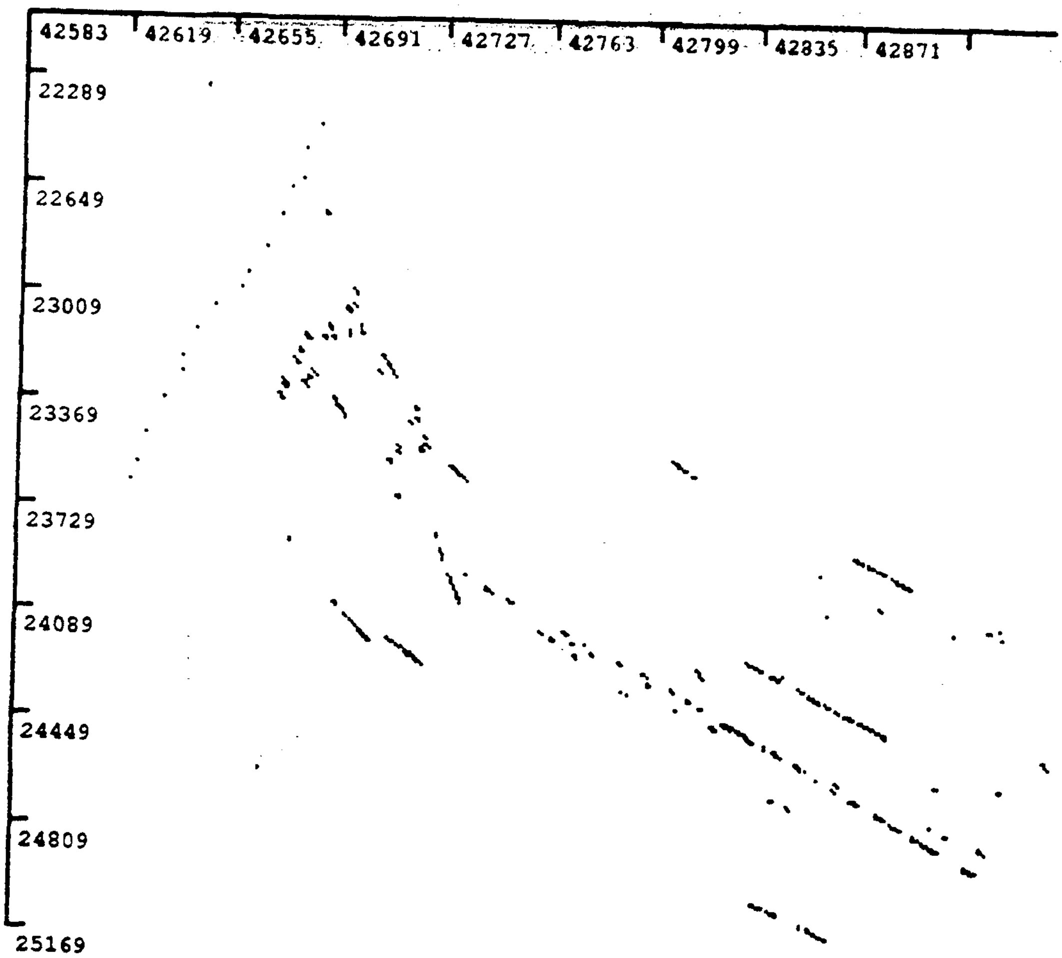

Figure A-103. Input data from data set 33.

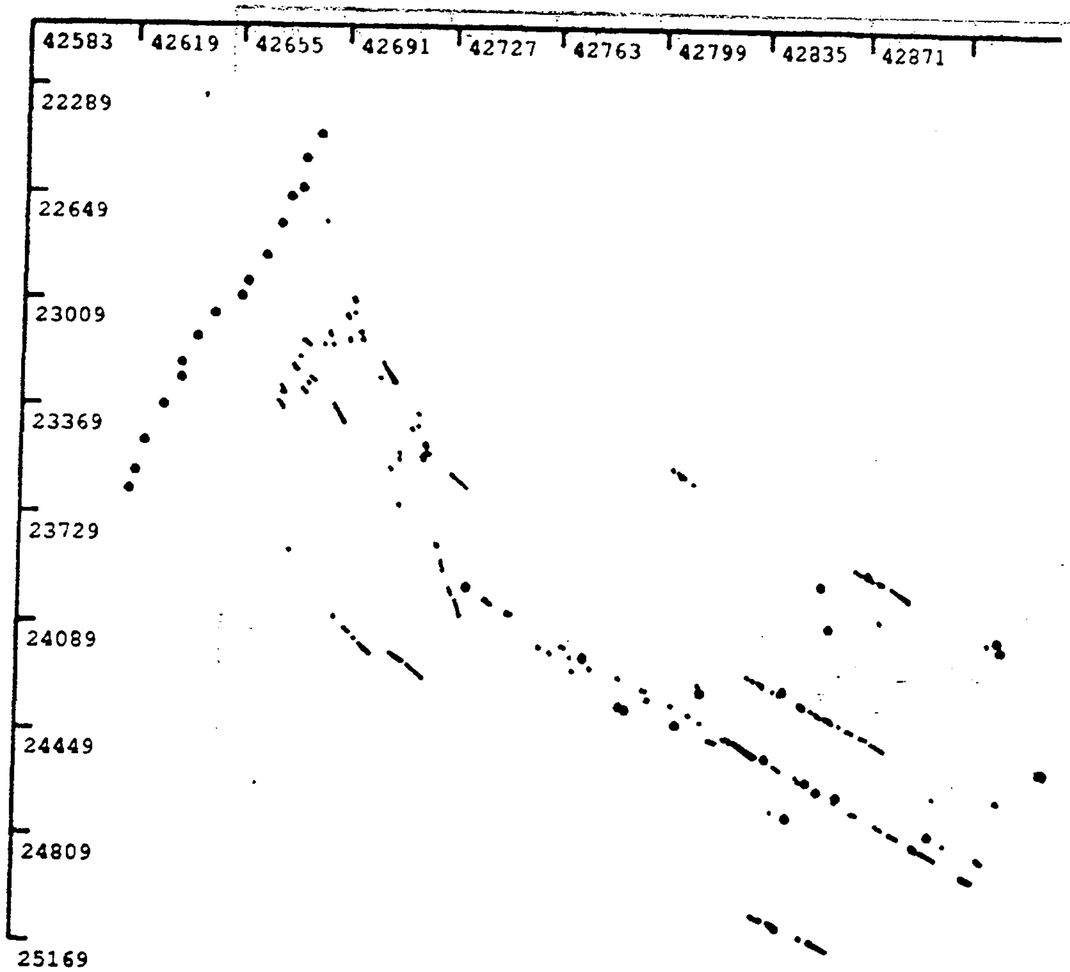

Figure A-104. Initial clusters from data set 33.

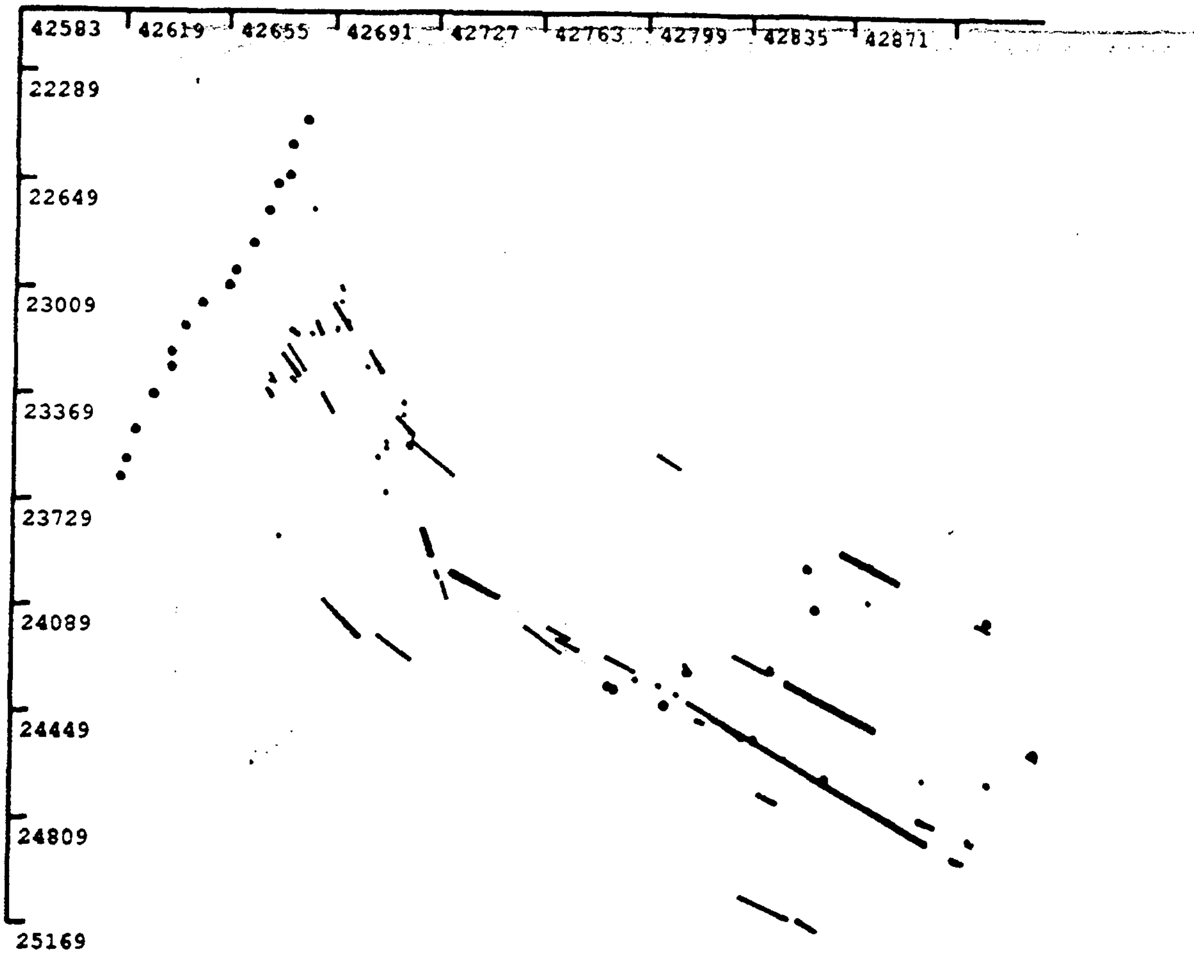

Figure A-105. Linear and online clusters from data set 33.

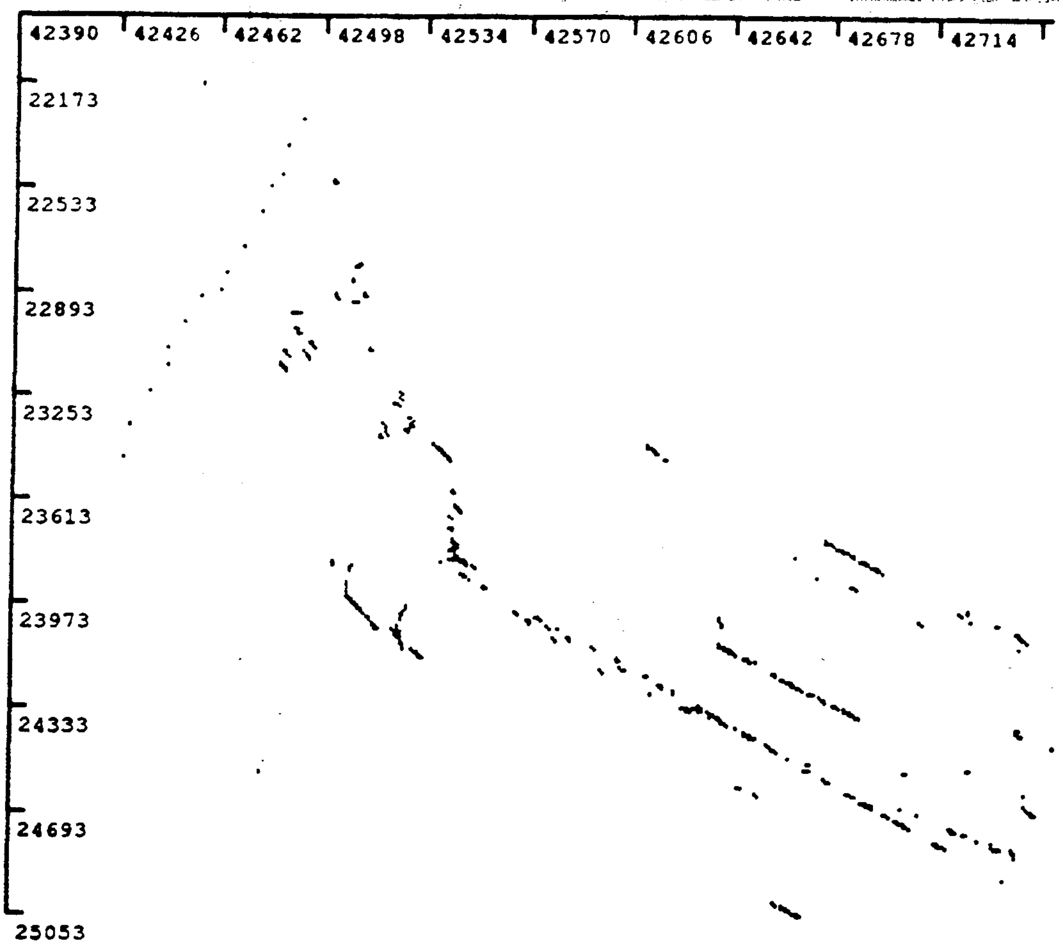

Figure A-106. Input data from data set 34.

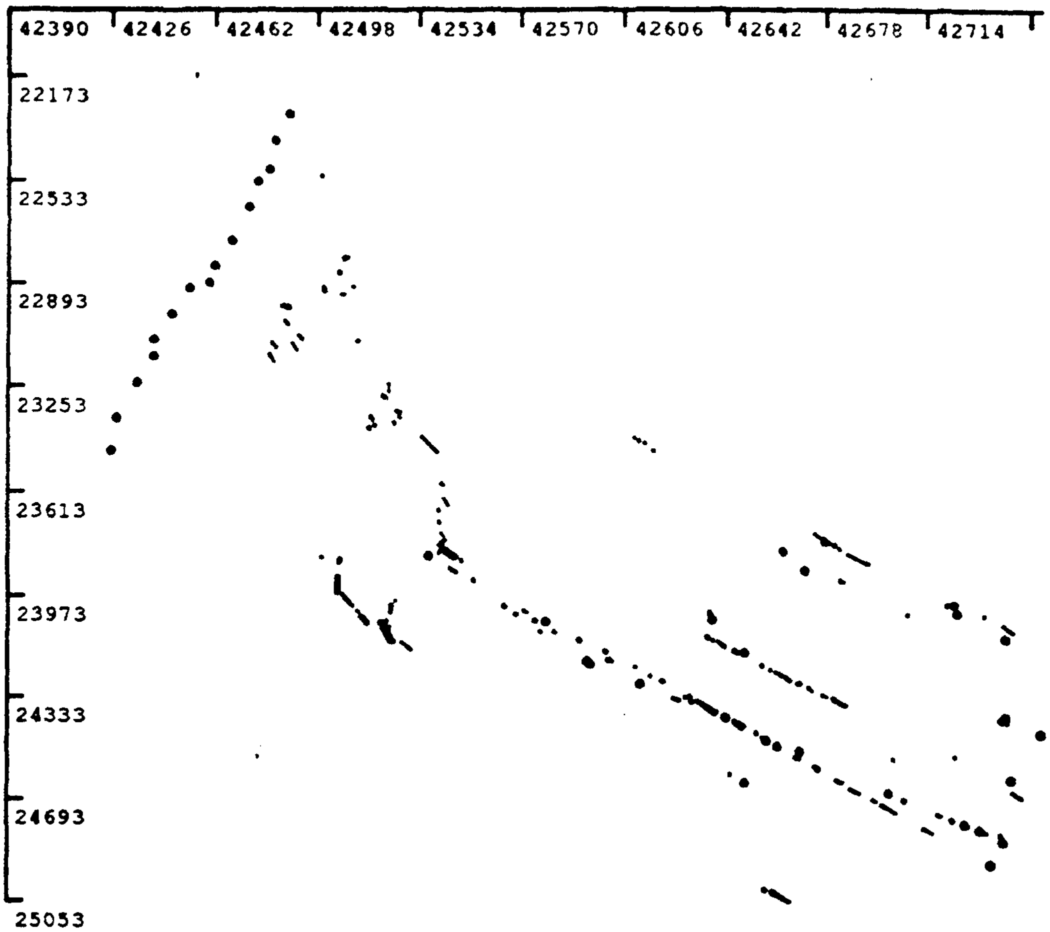

Figure A-107. Initial clusters from data set 34.

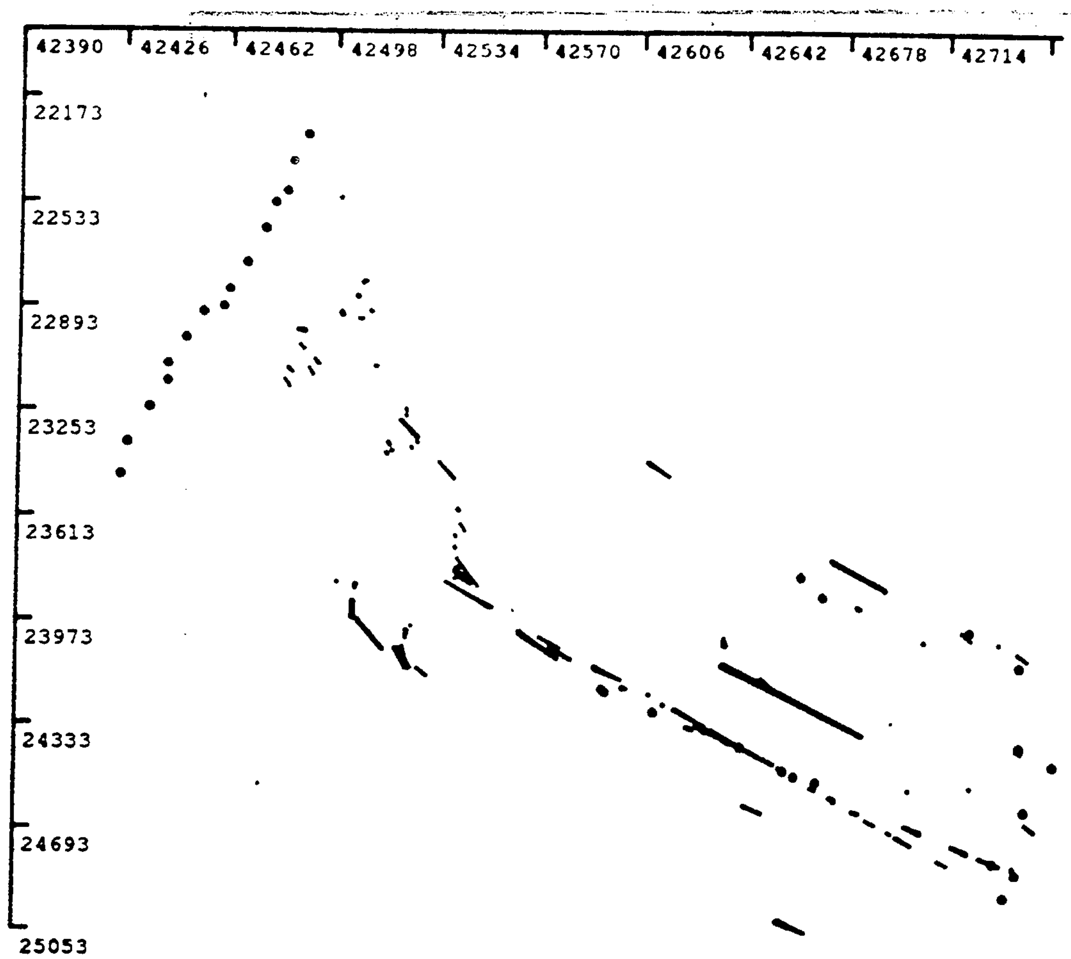

Figure A-108. Linear and online clusters from data set 34.

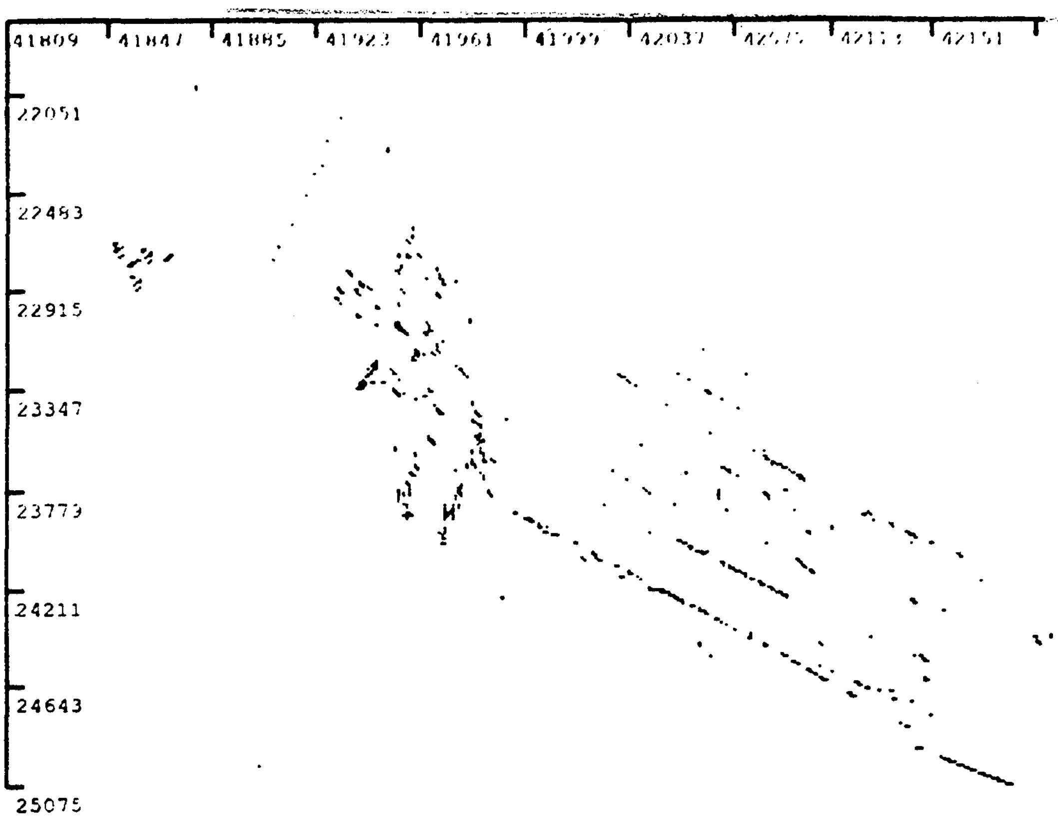

Figure A-109. Input data from data set 35.

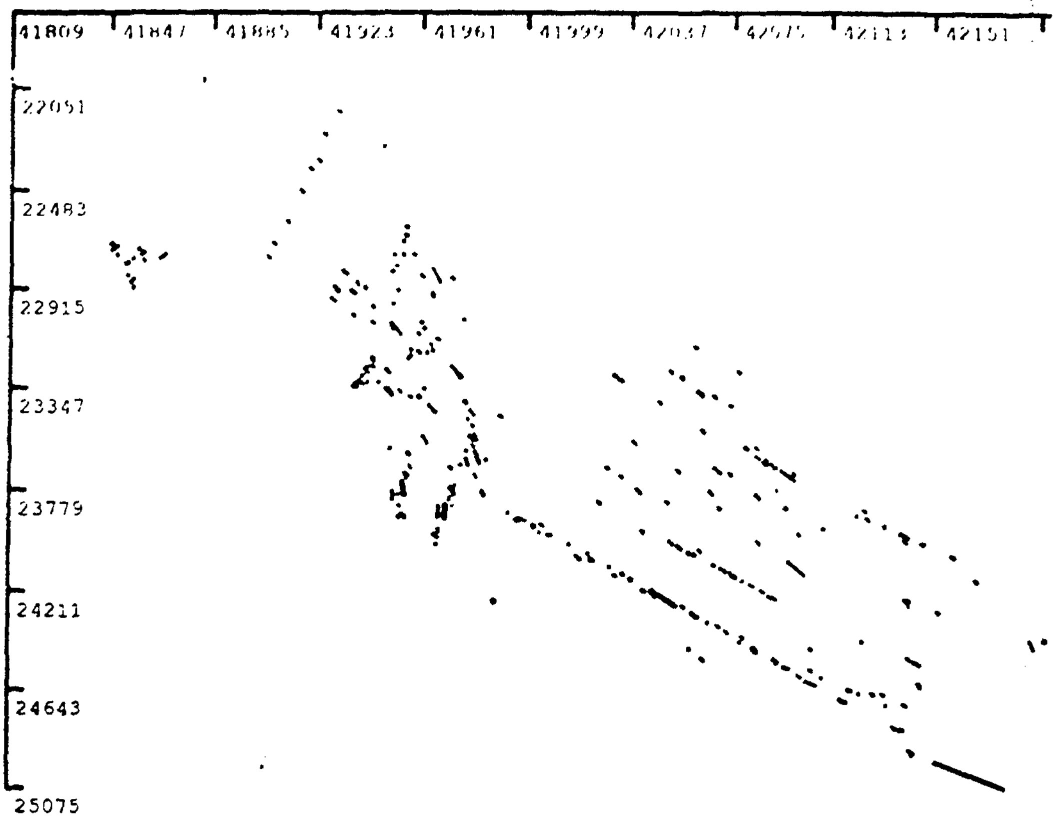

Figure A-110. Initial clusters from data set 35.

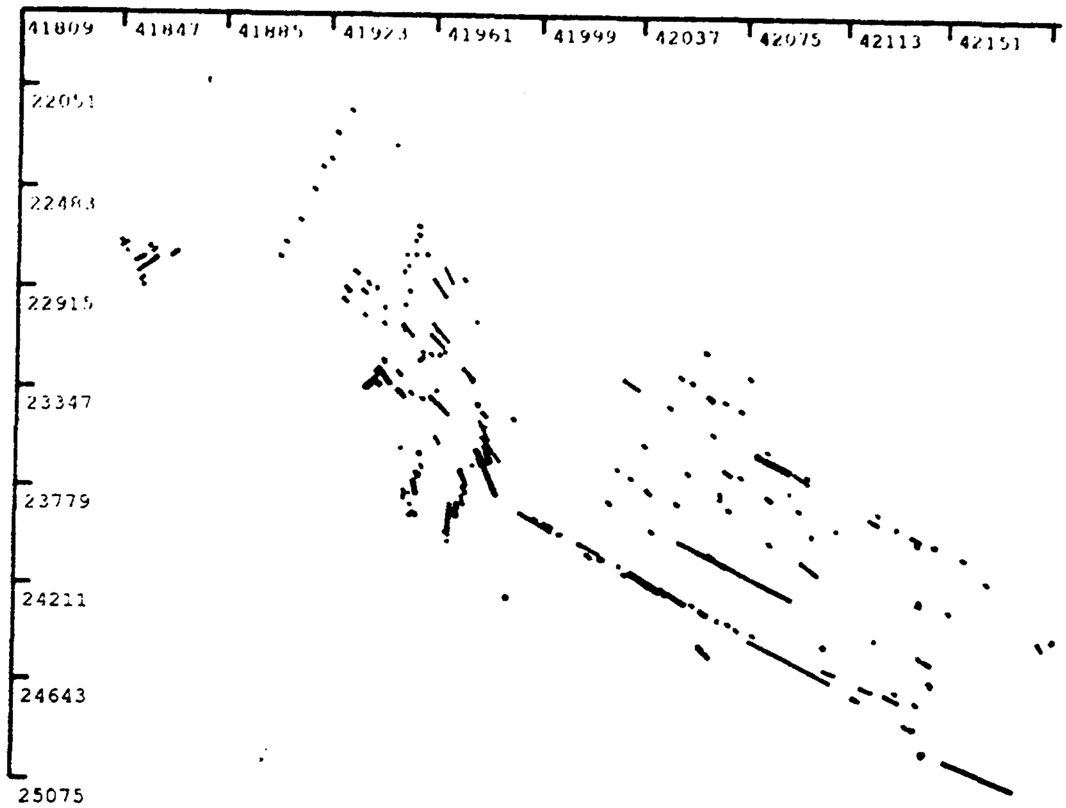

Figure A-111. Linear and online clusters from data set 35.

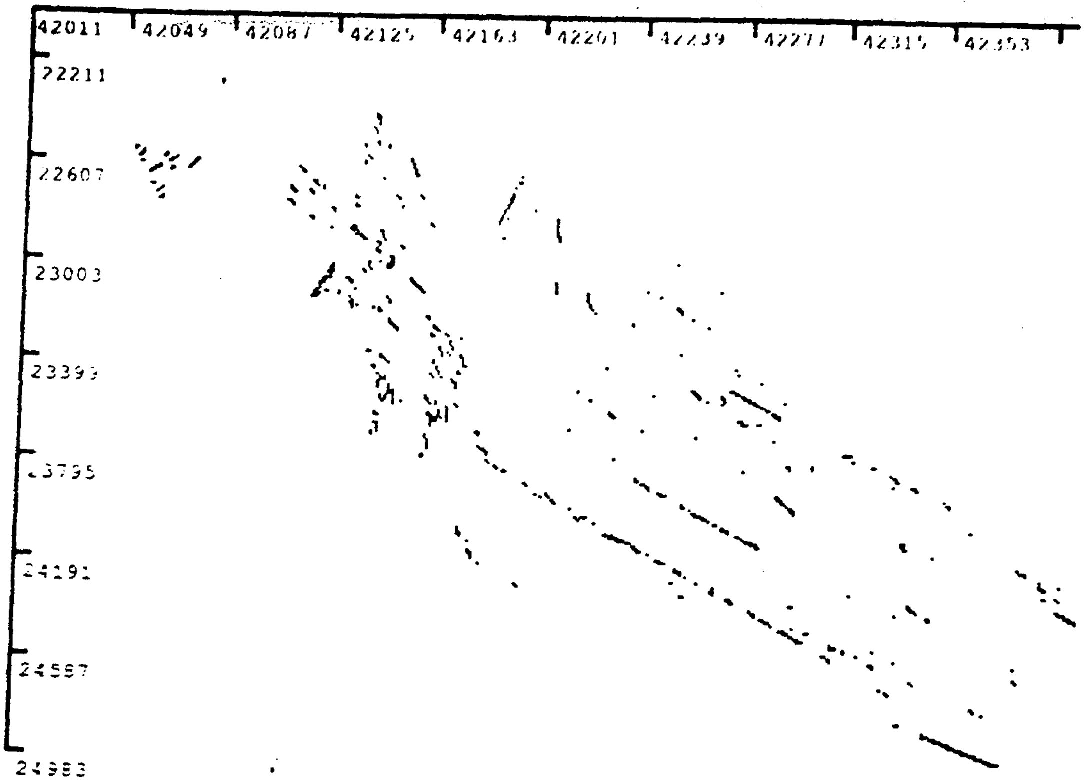

Figure A-112. Input data from data set 36.

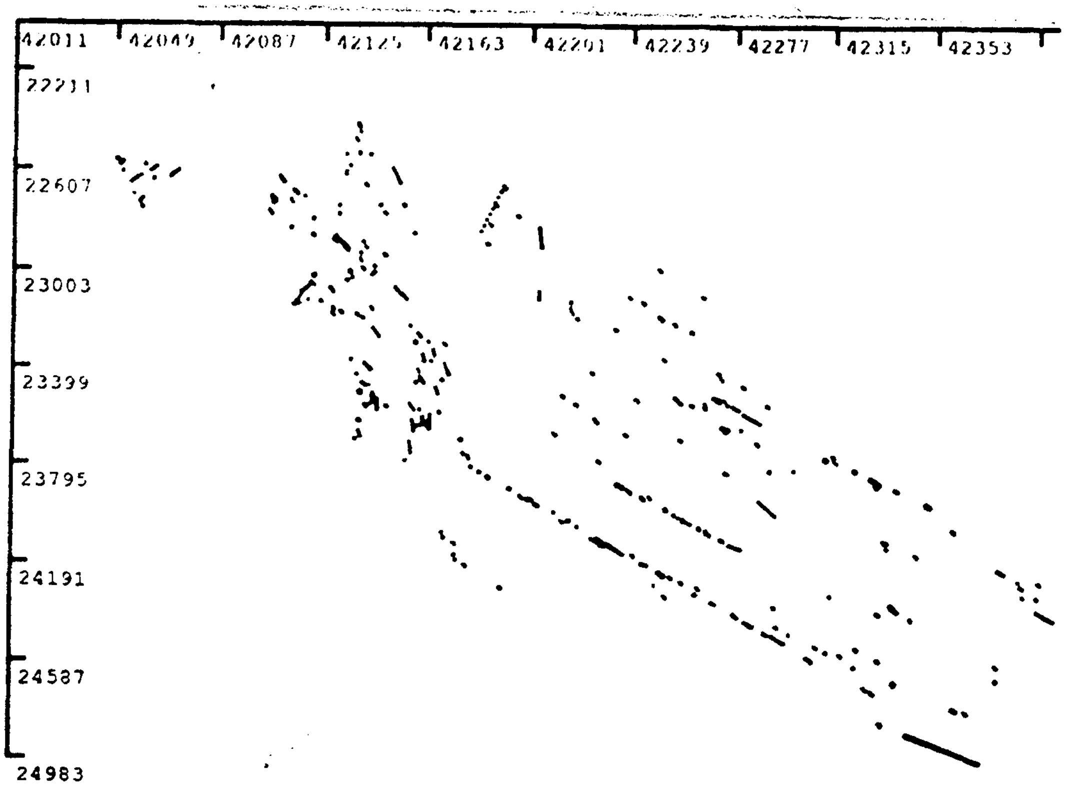

Figure A-113. Initial clusters from data set 36.

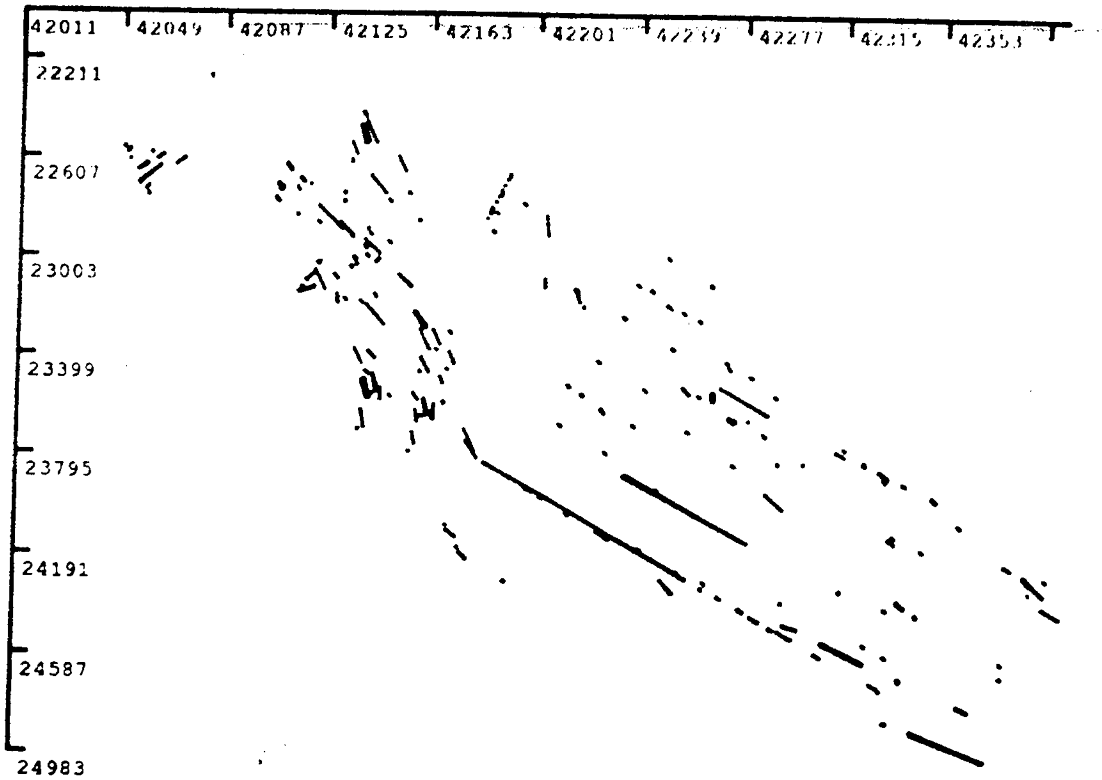

Figure A-114. Linear and online clusters from data set 36.

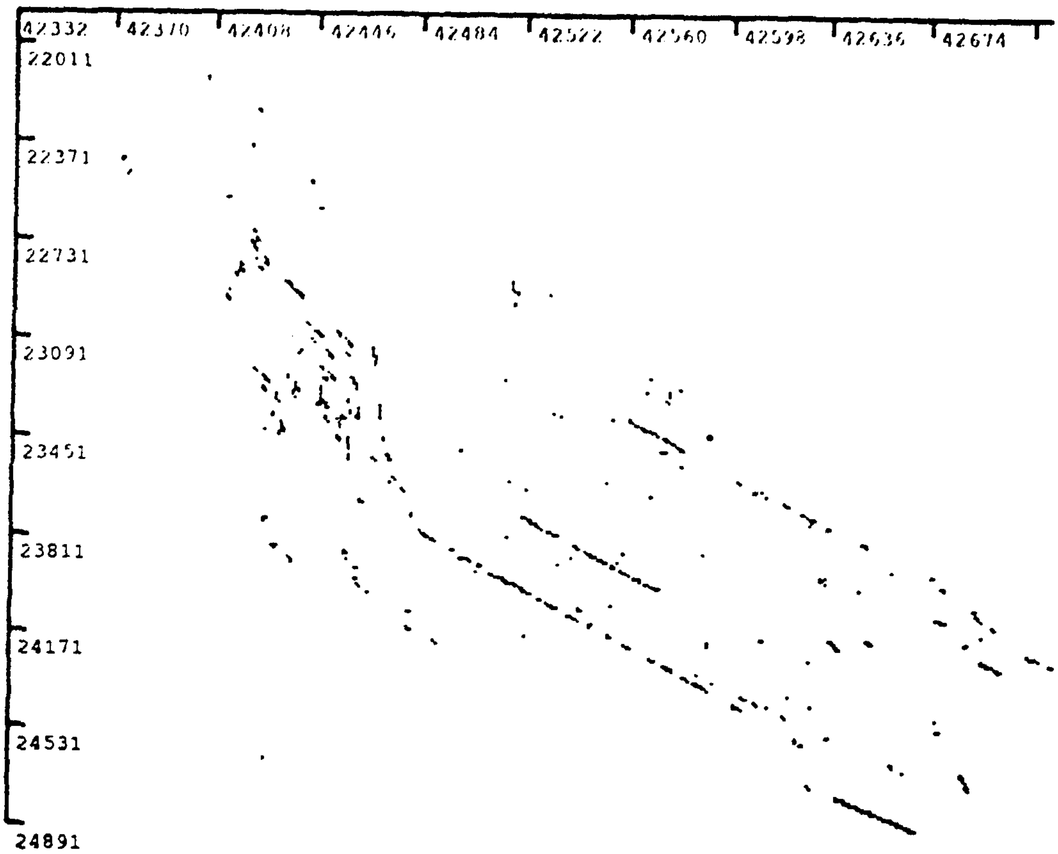

Figure A-115. Input data from data set 37.

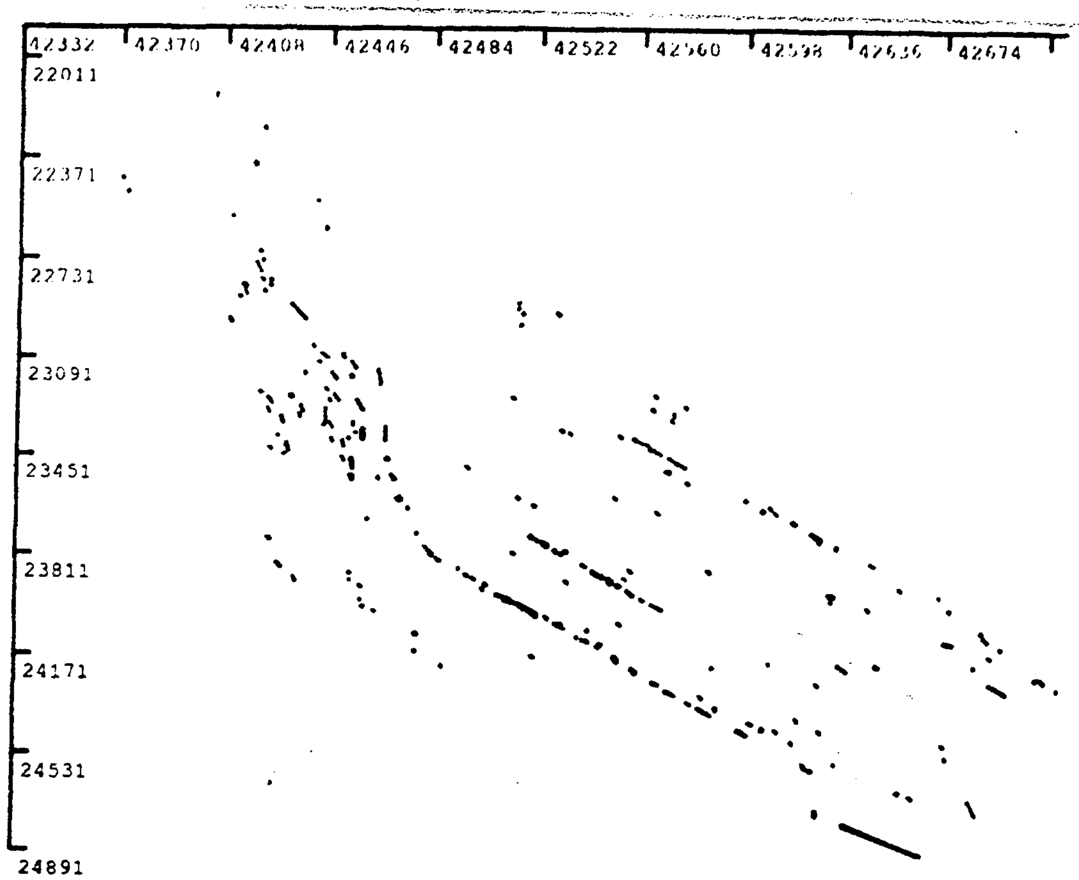

Figure A-116. Initial clusters from data set 37.

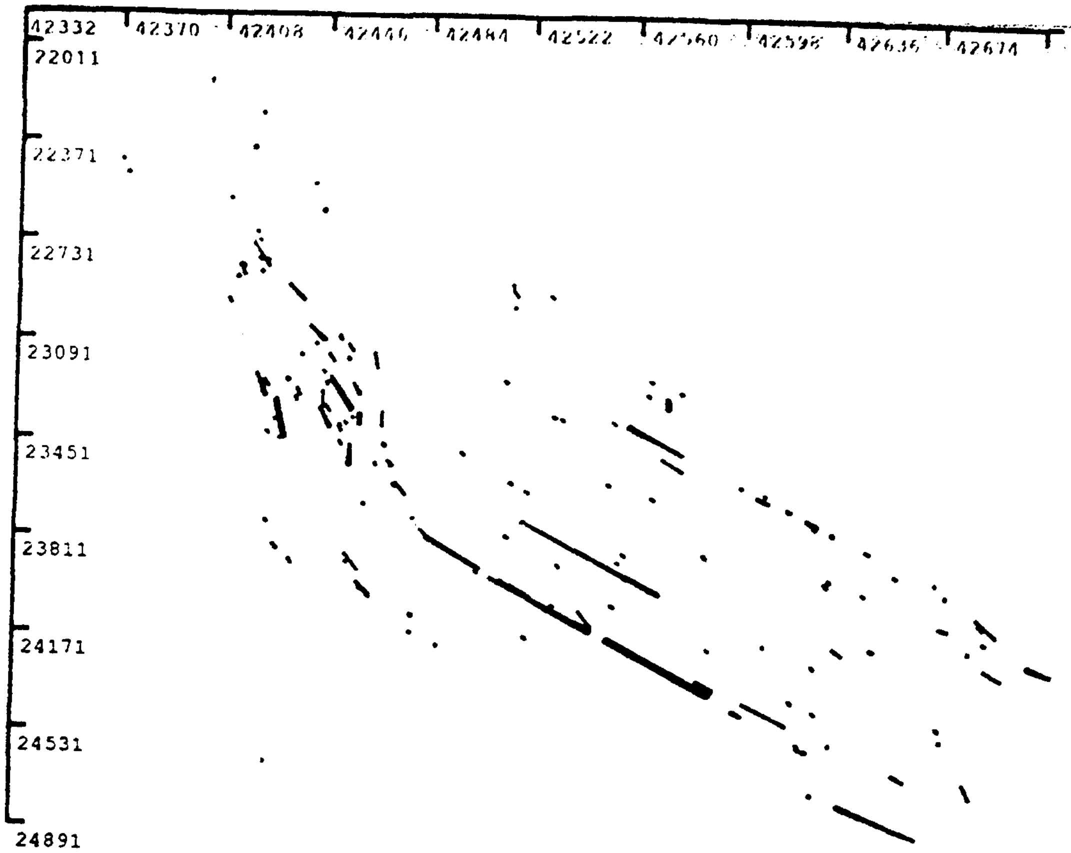

Figure A-117. Linear and online clusters from data set 37.

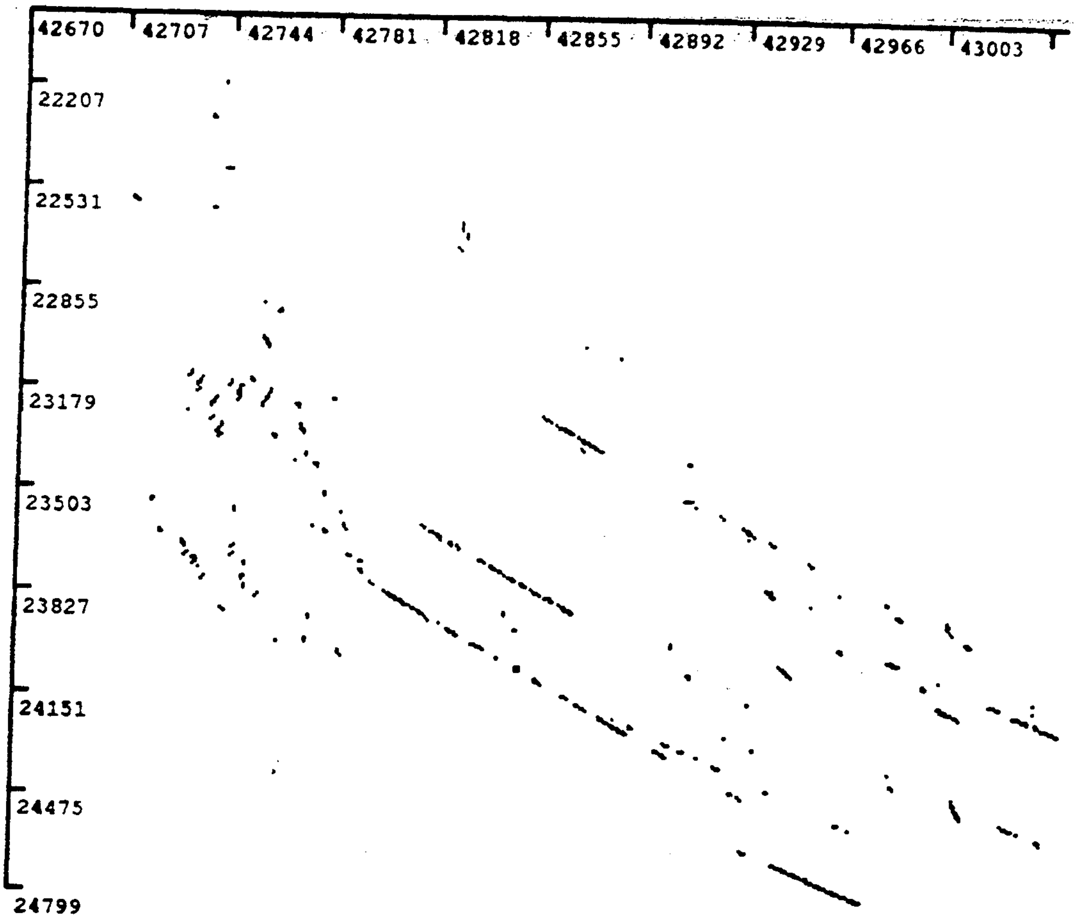

Figure A-118. Input data from data set 38.

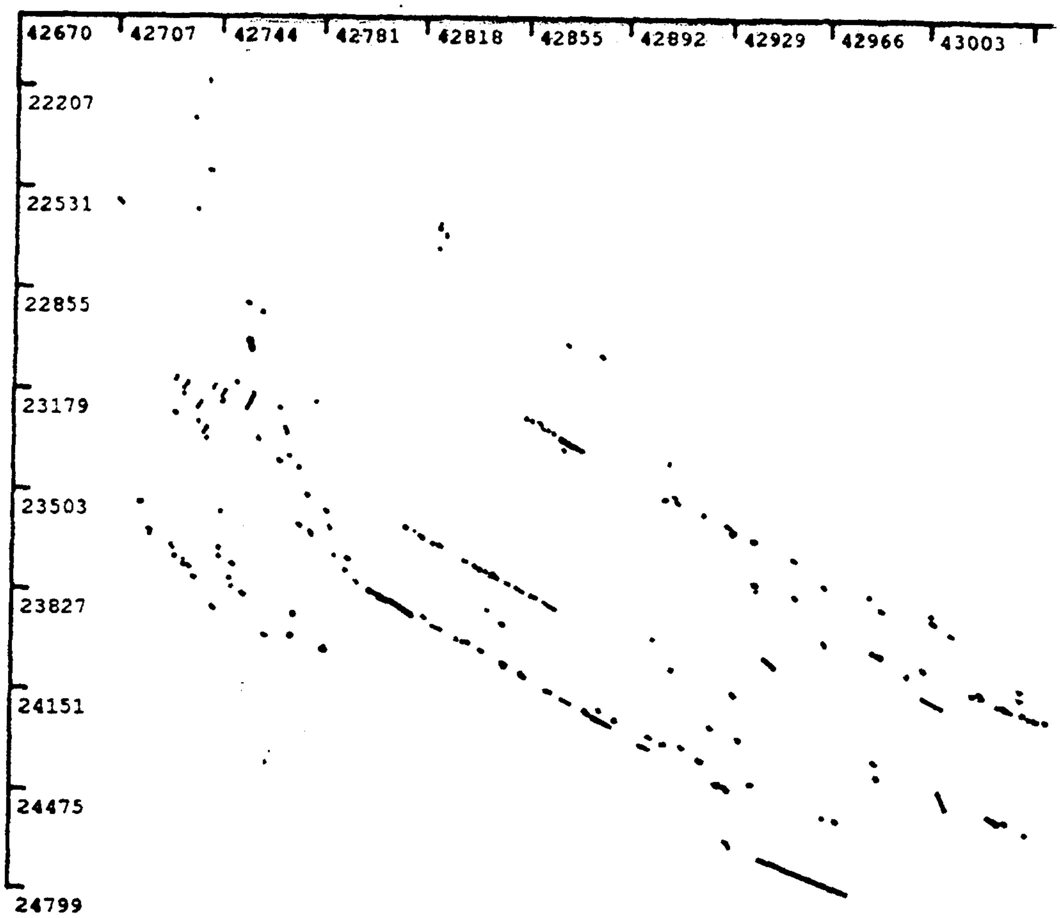

Figure A-119. Initial clusters from data set 38.

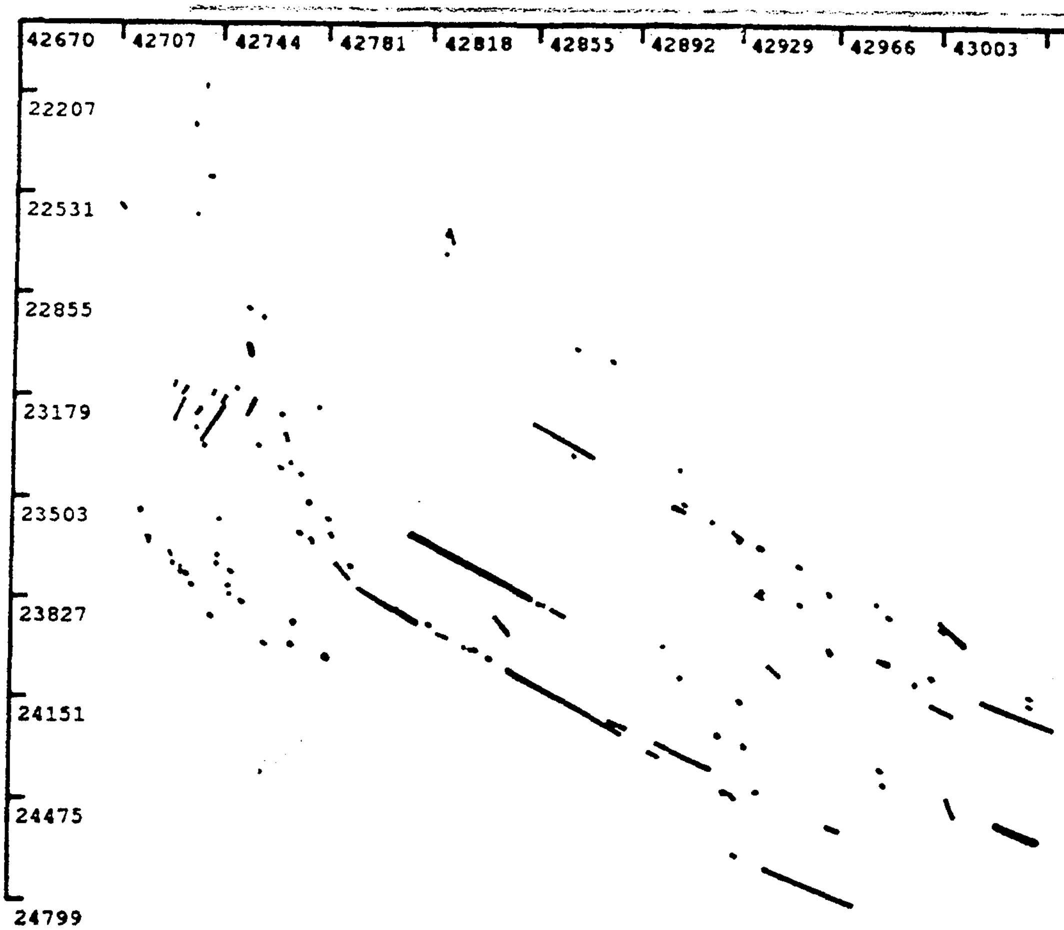

Figure A-120. Linear and online clusters from data set 38.

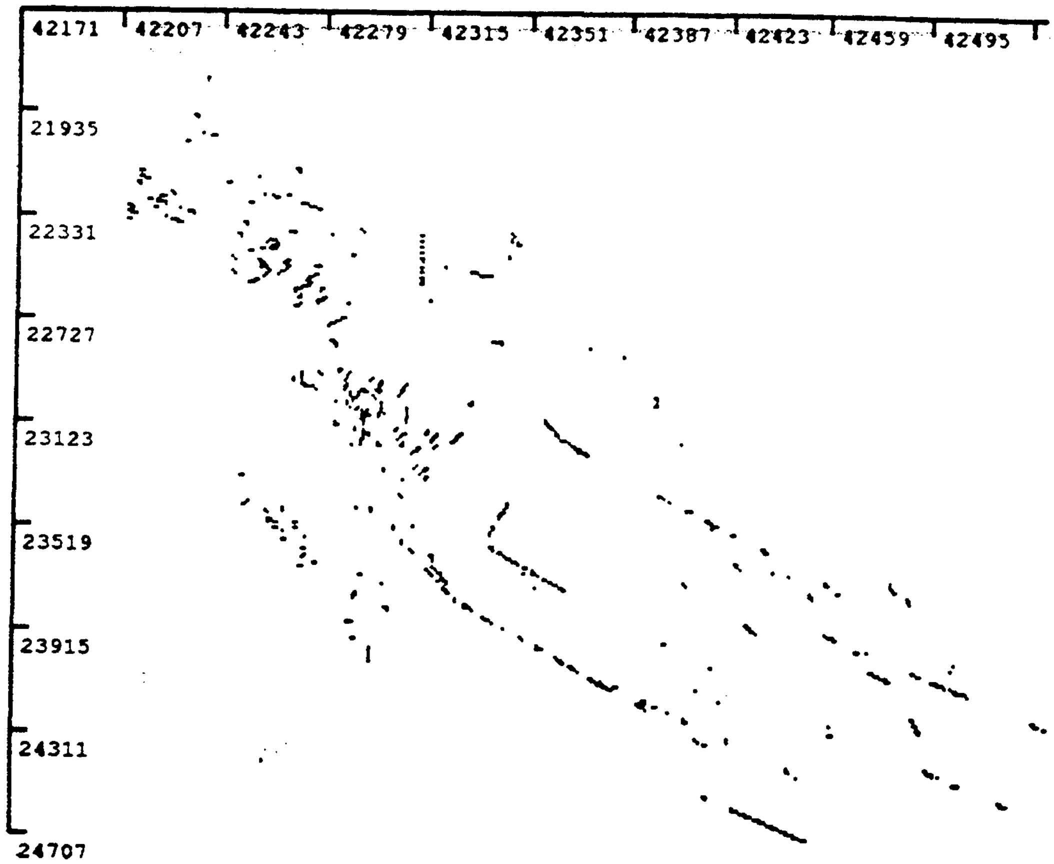

Figure A-121. Input data from data set 39.

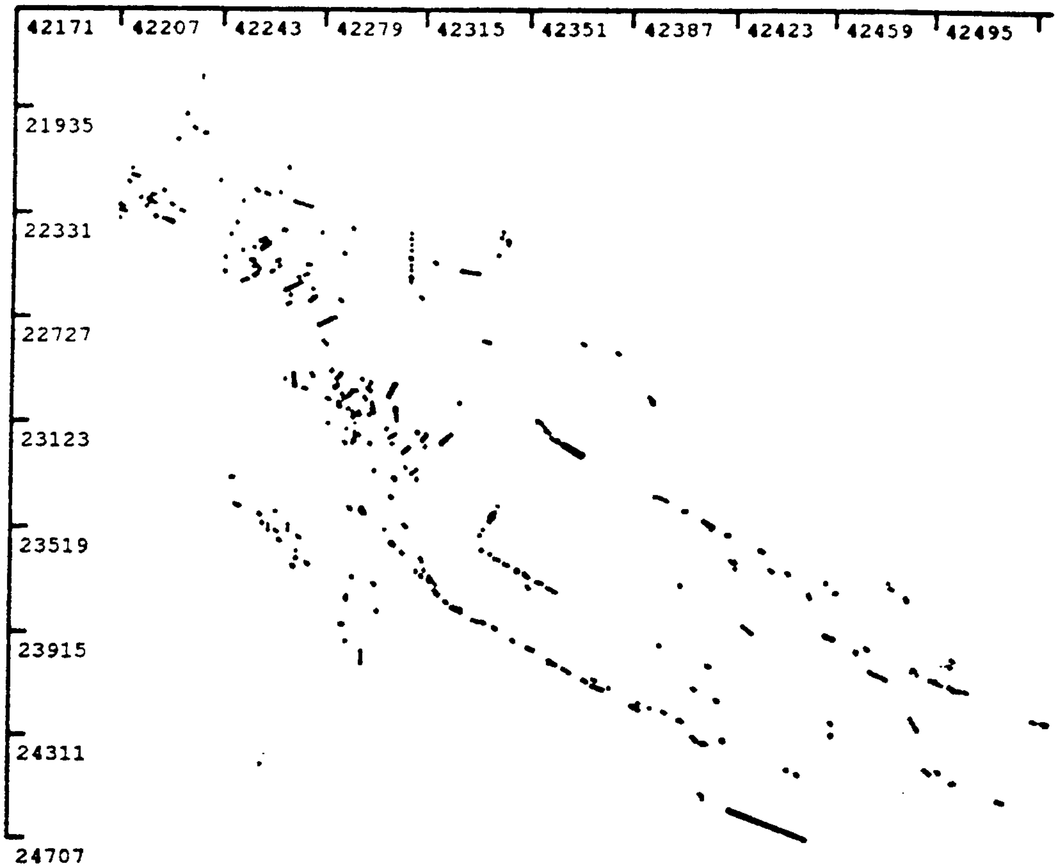

Figure A-122. Initial clusters from data set 39.

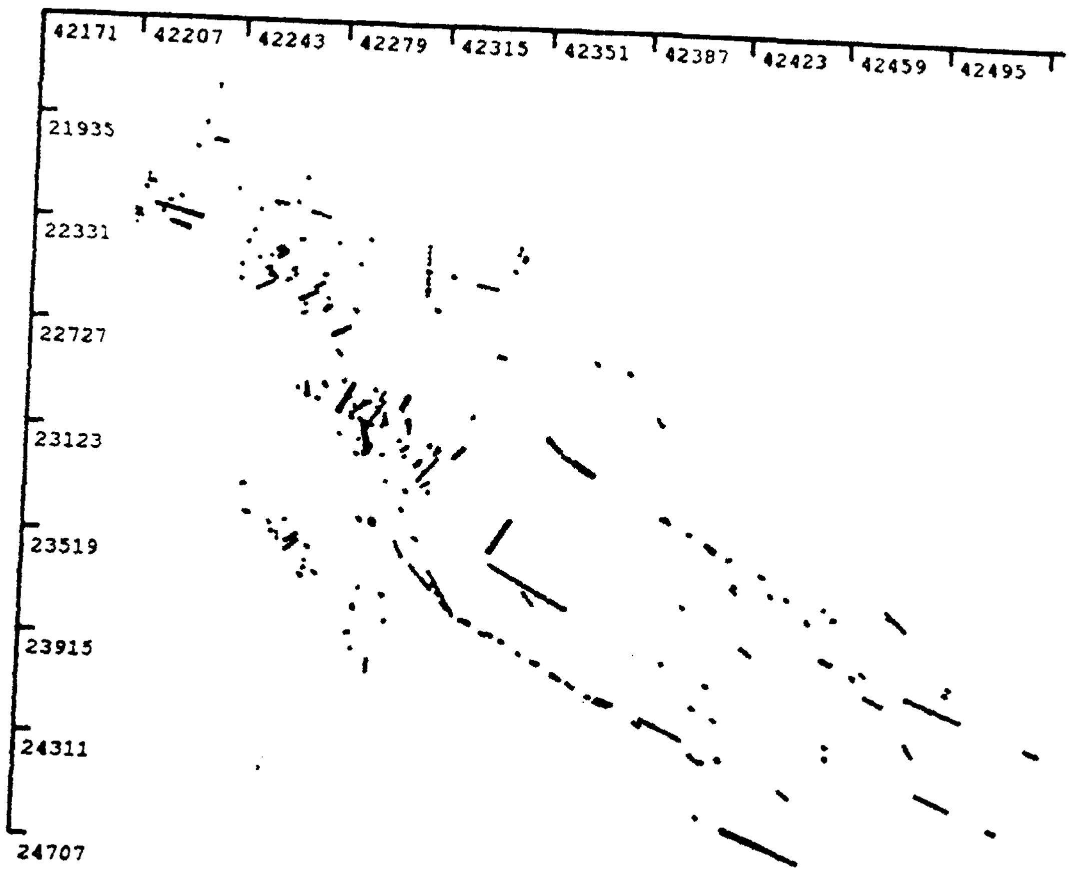

Figure A-123. Linear and online clusters from data set 39.

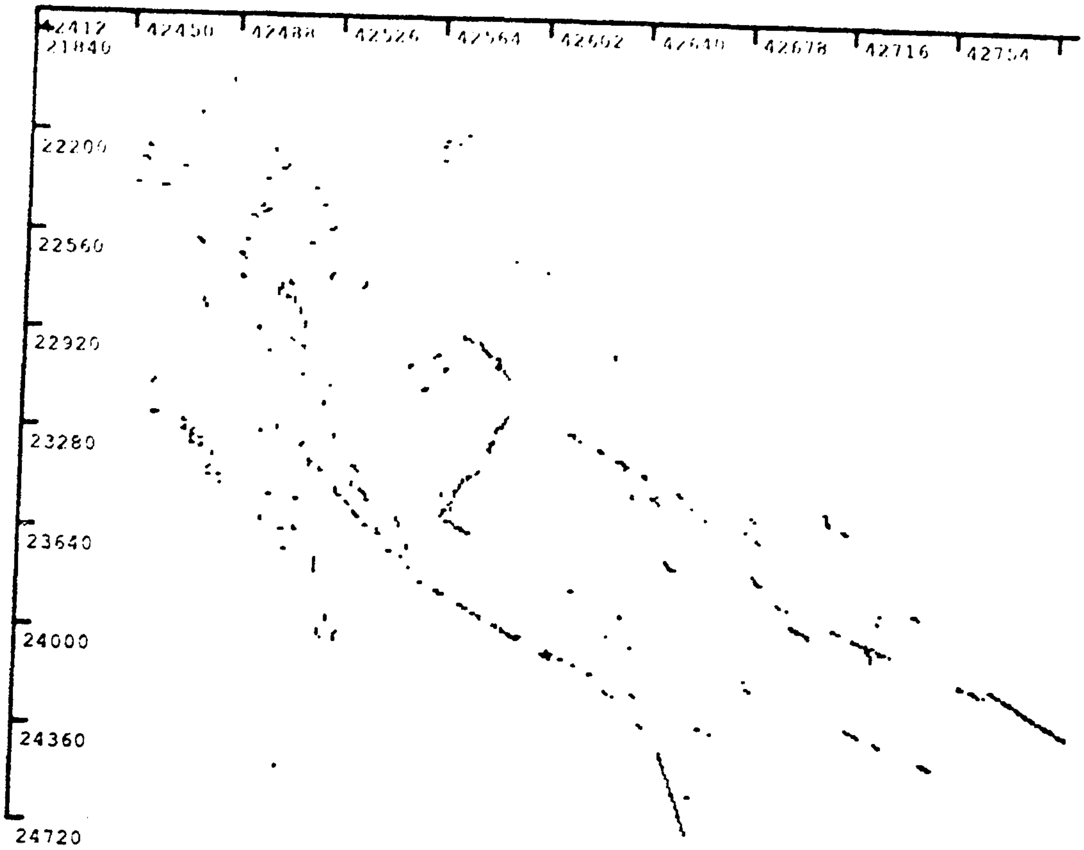

Figure A-124. Input data from data set 40.

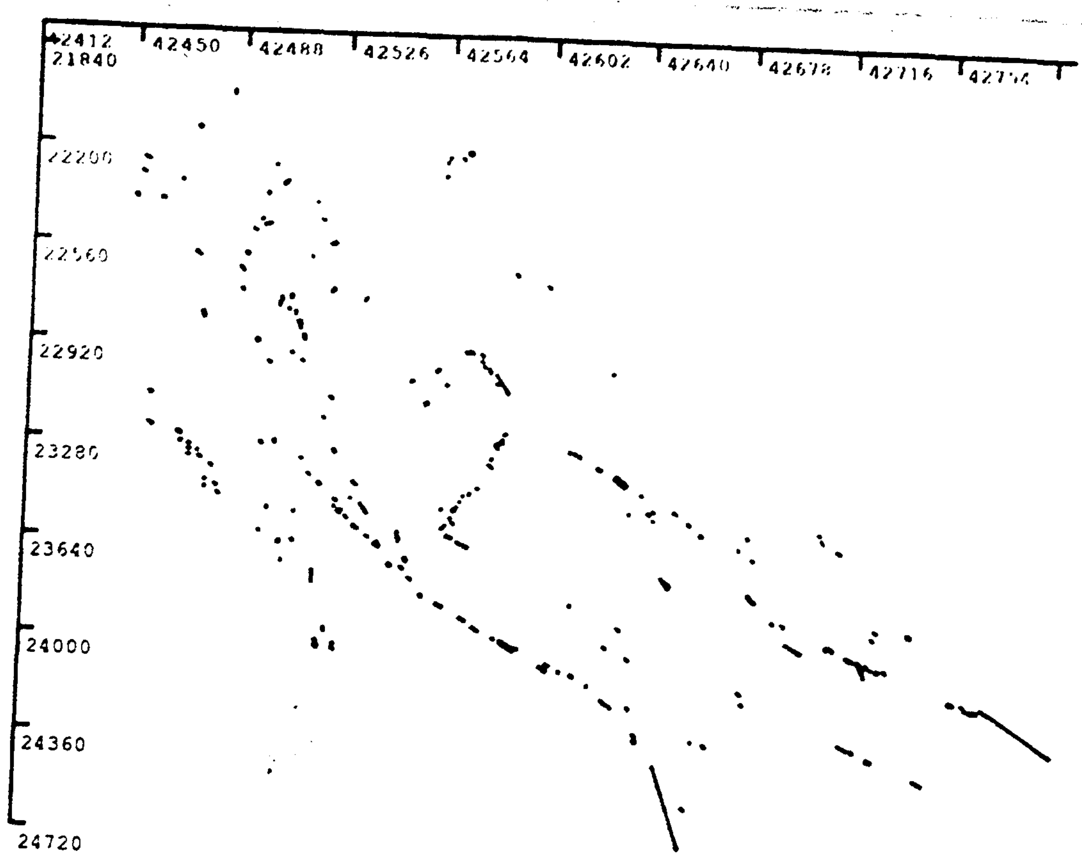

Figure A-125. Initial clusters from data set 40.

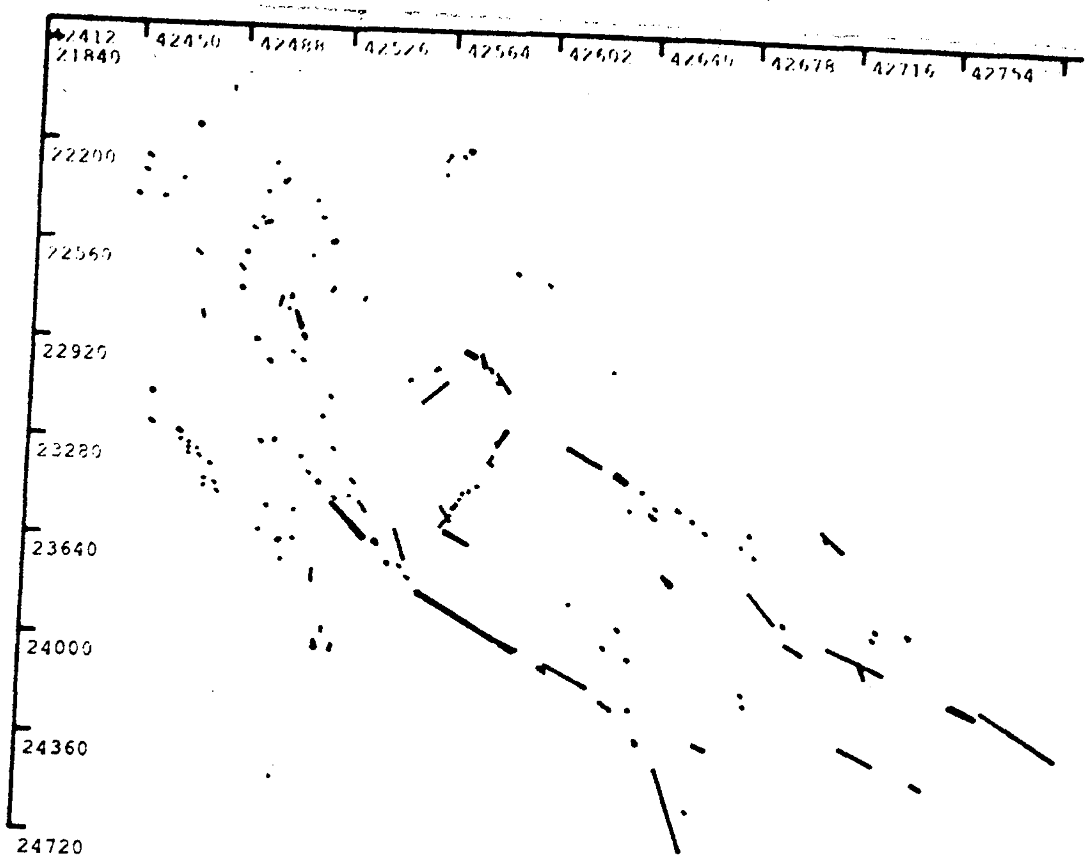

Figure A-126. Linear and online clusters from data set 40.

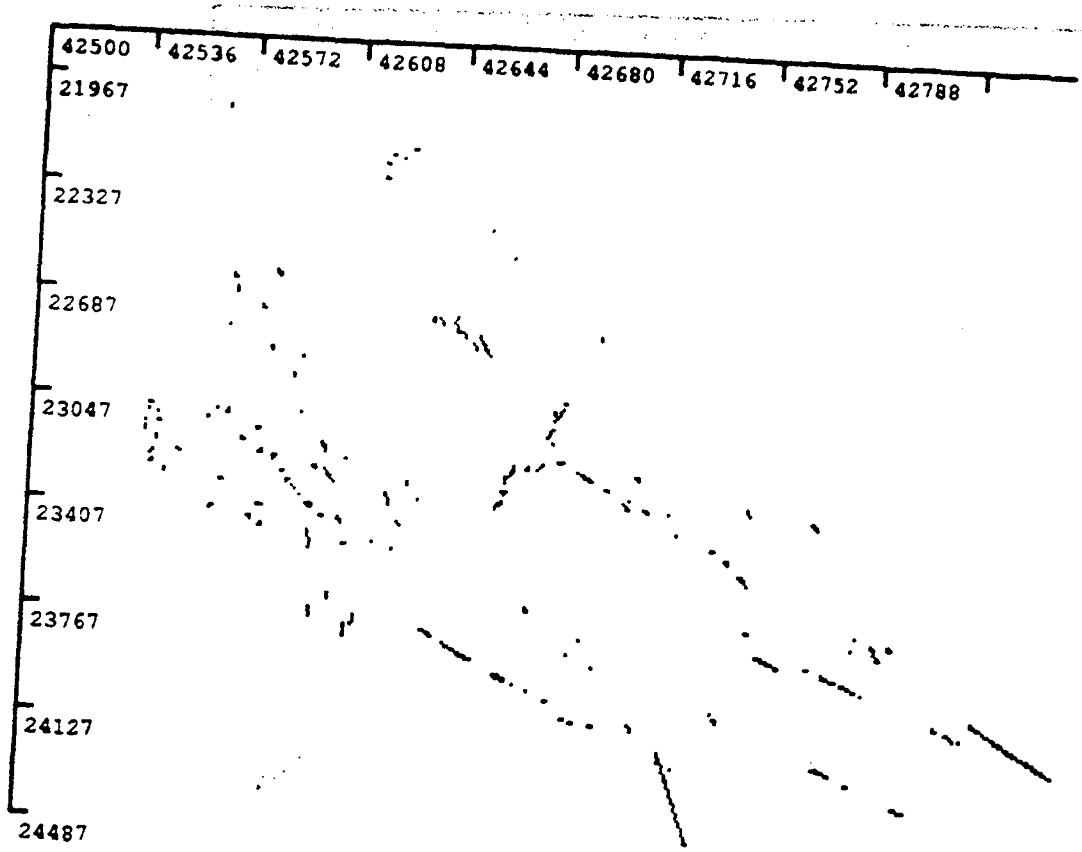

Figure A-127. Input data from data set 41.

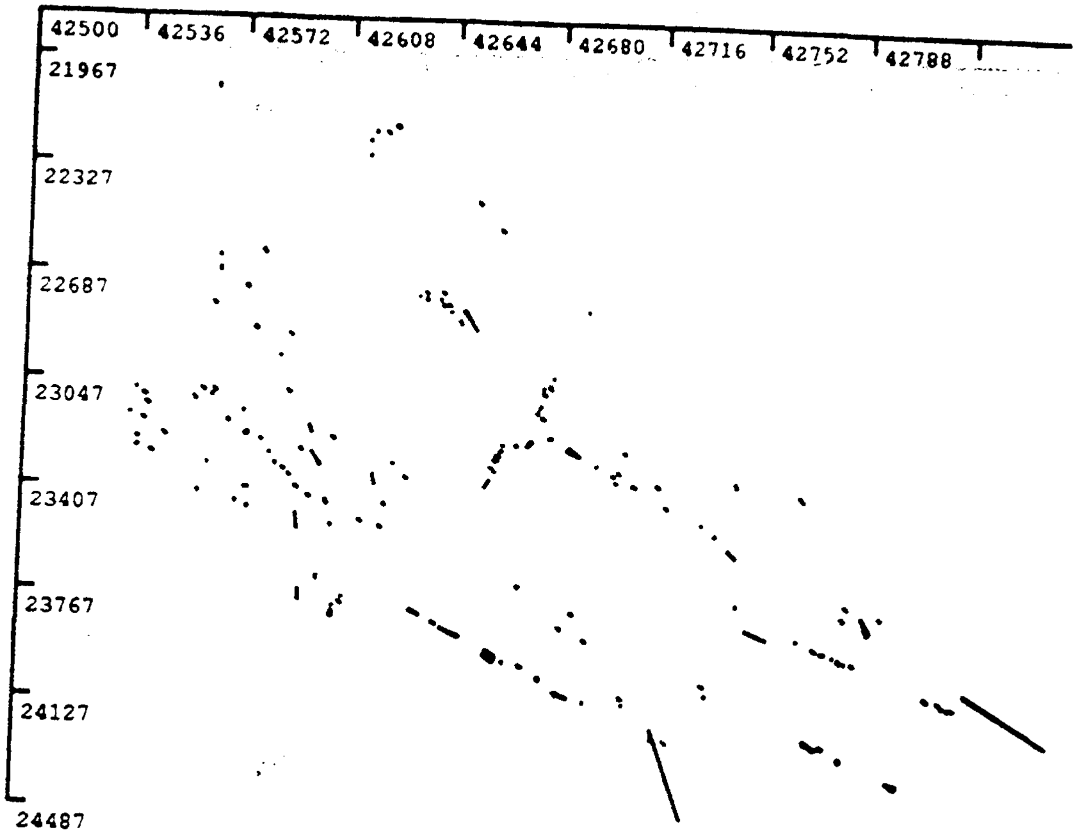

Figure A-128. Initial clusters from data set 41.

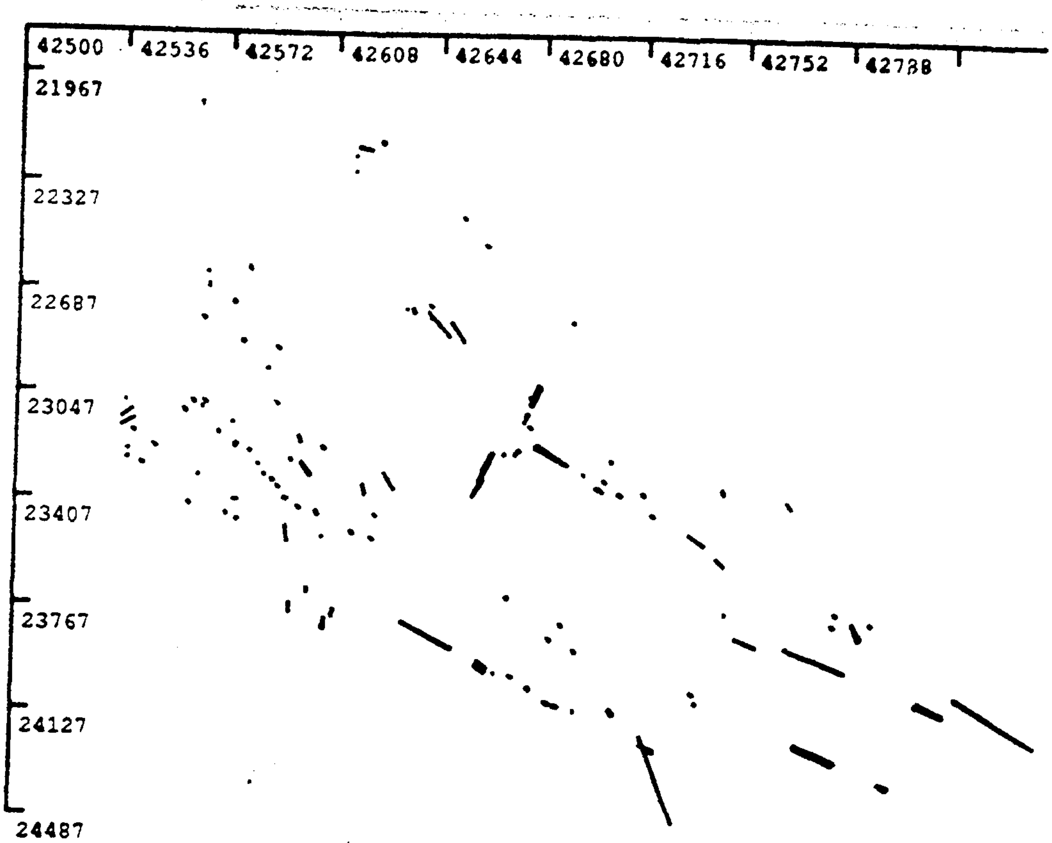

Figure A-129. Linear and online clusters from data set 41.

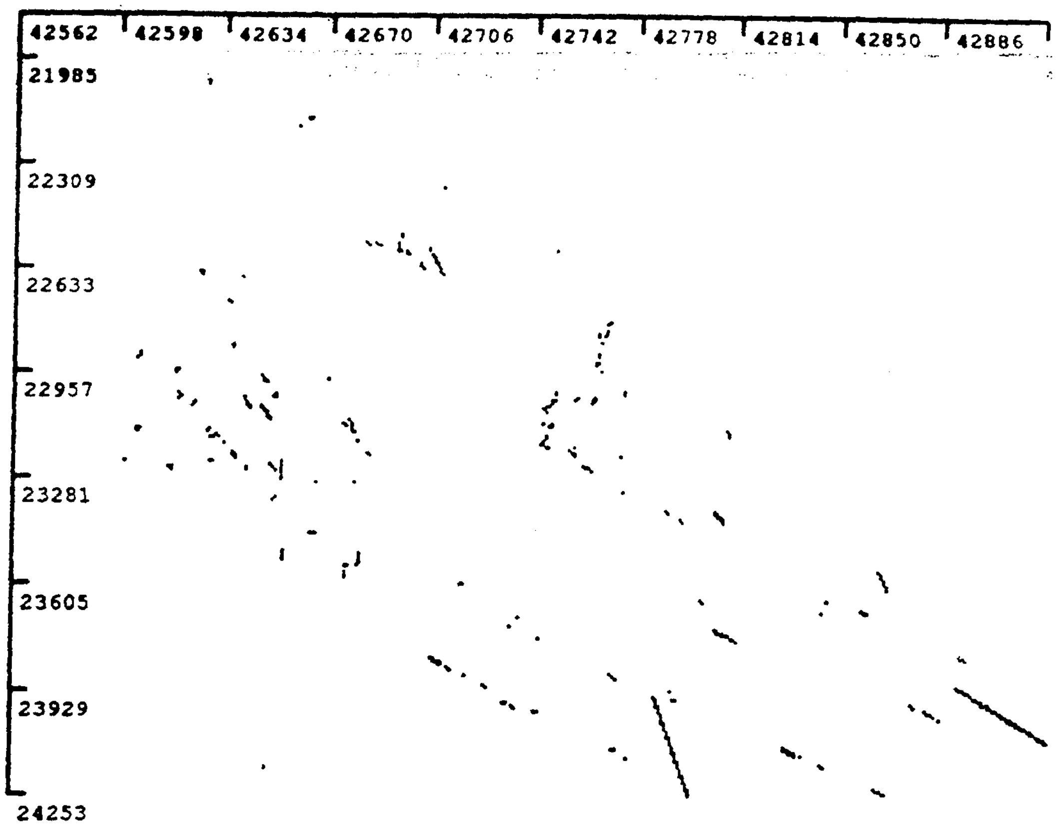

Figure A-130. Input data from data set 42.

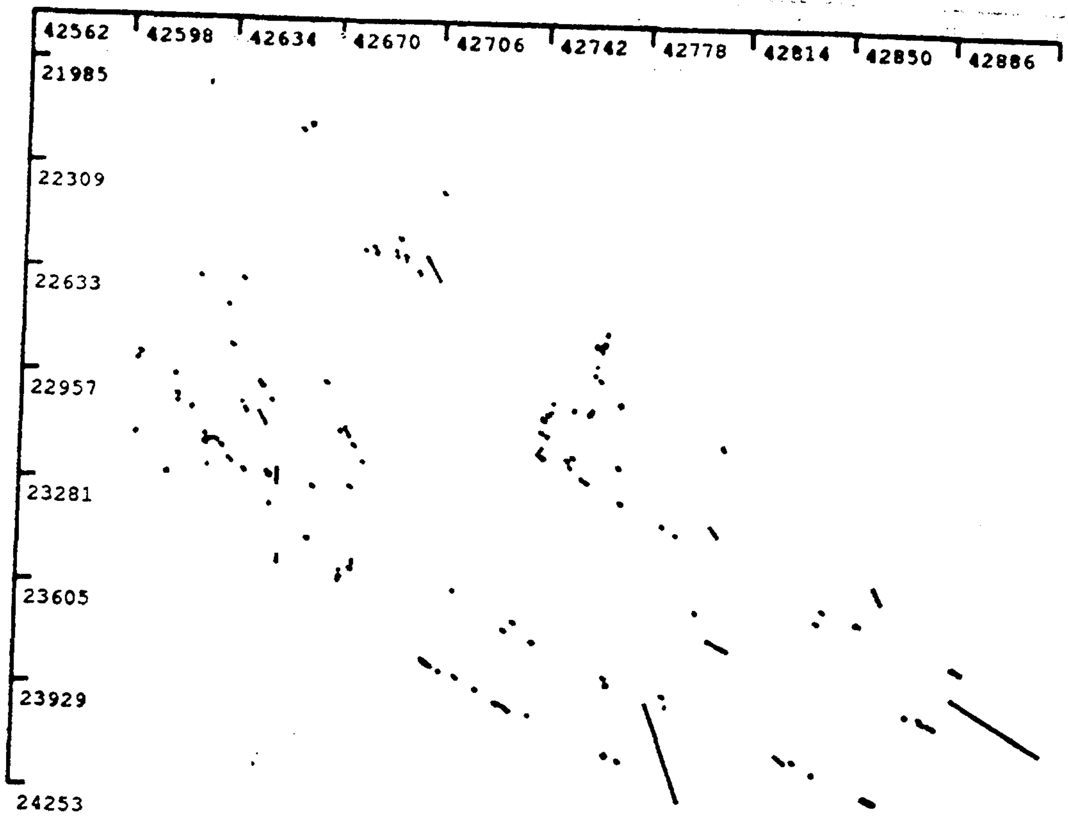

Figure A-131. Initial clusters from data set 42.

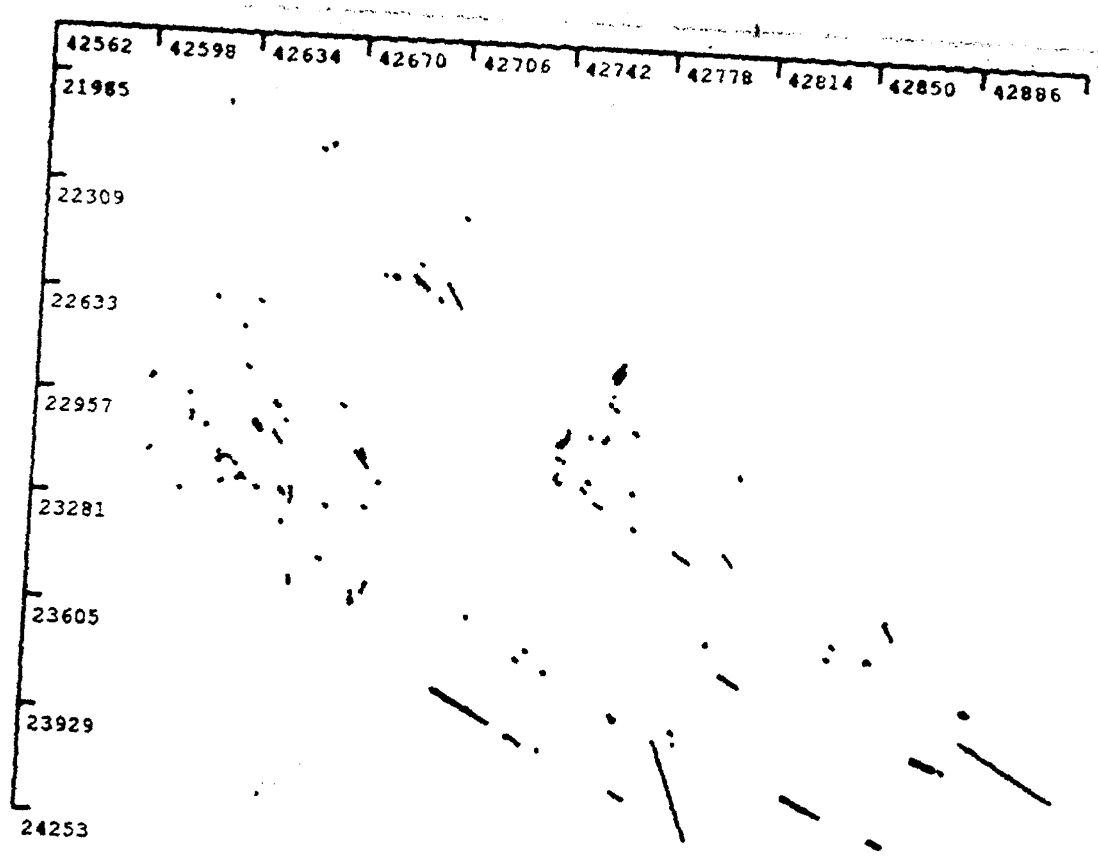

Figure A-132. Linear and online clusters from data set 42.

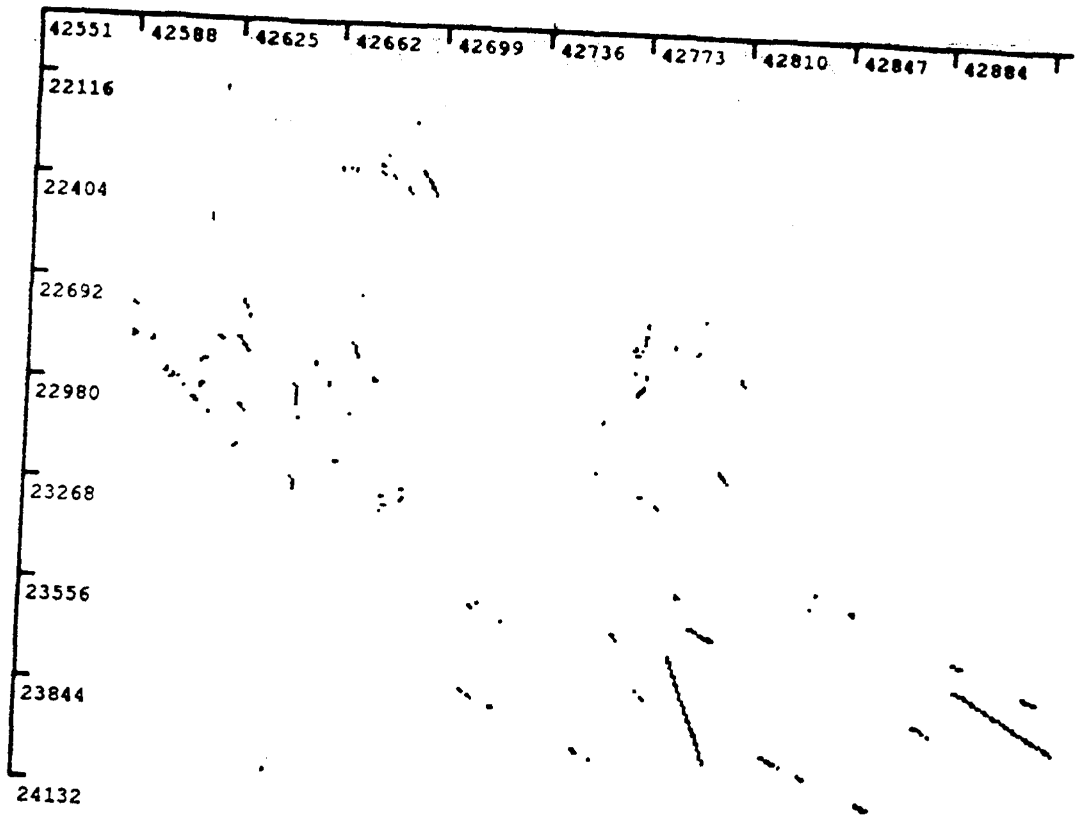

Figure A-133. Input data from data set 43.

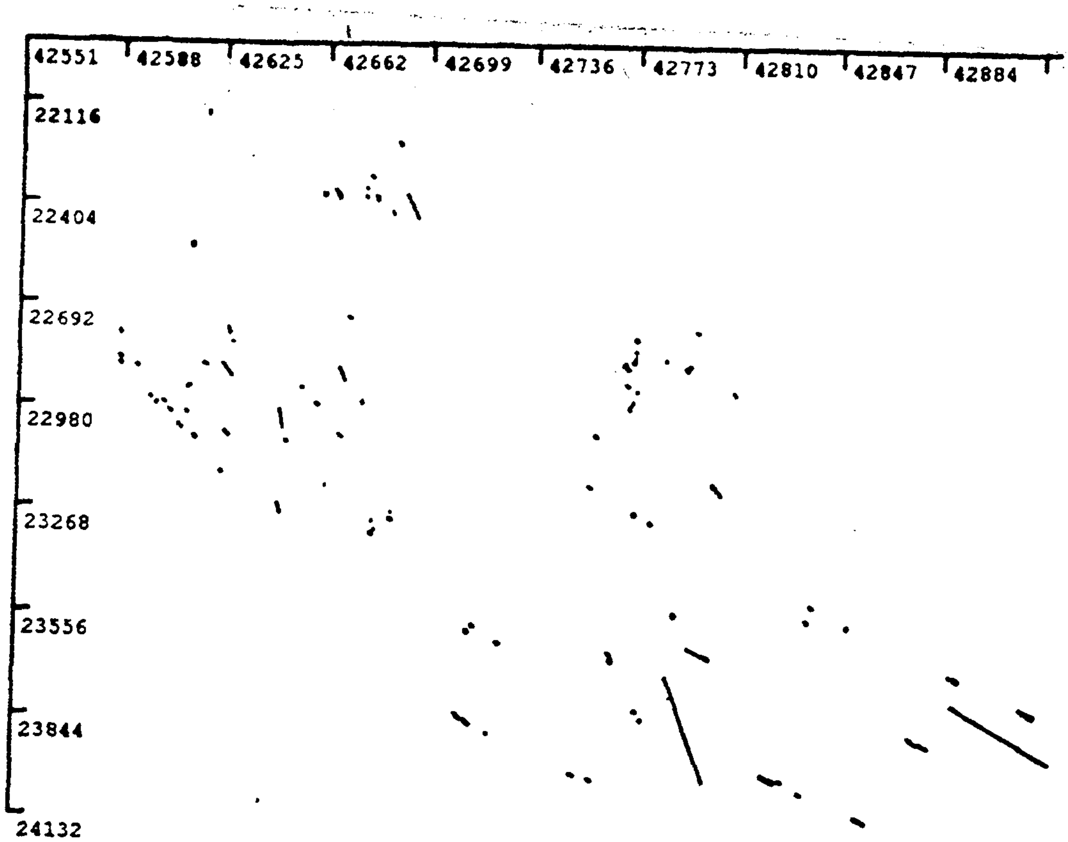

Figure A-134. Initial clusters from data set 43.

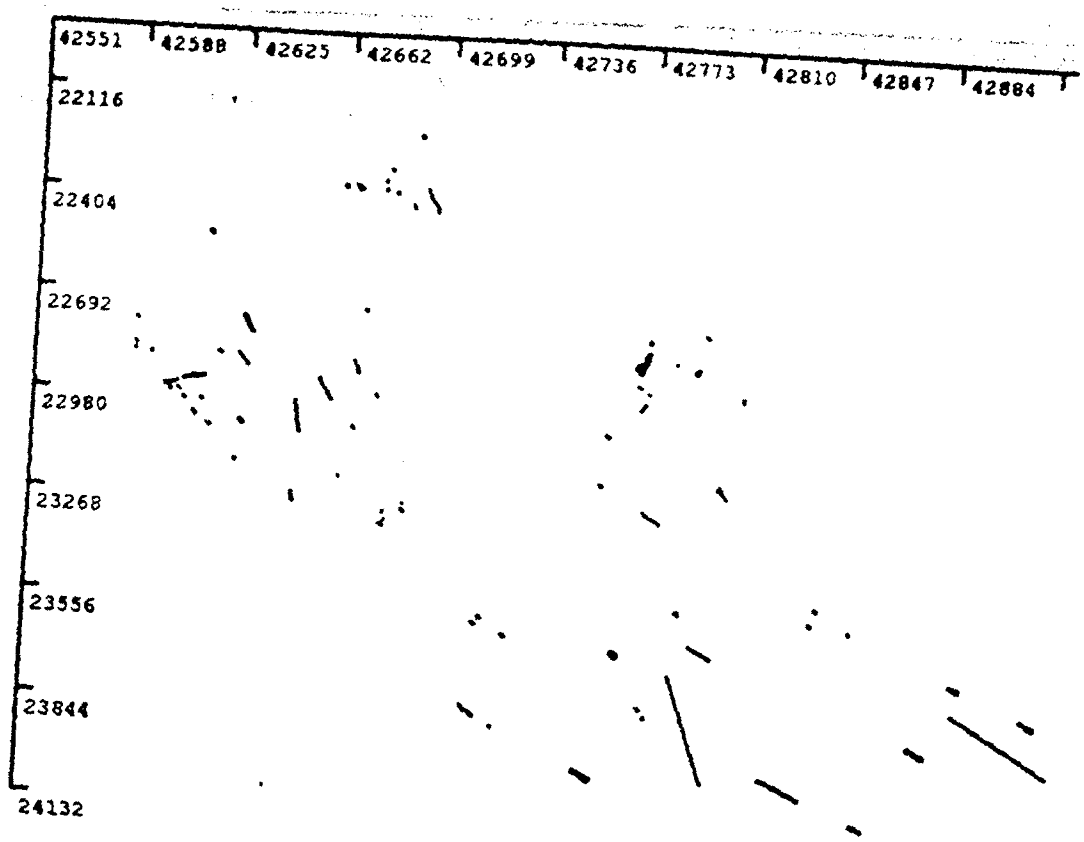

Figure A-135. Linear and online clusters from data set 43.

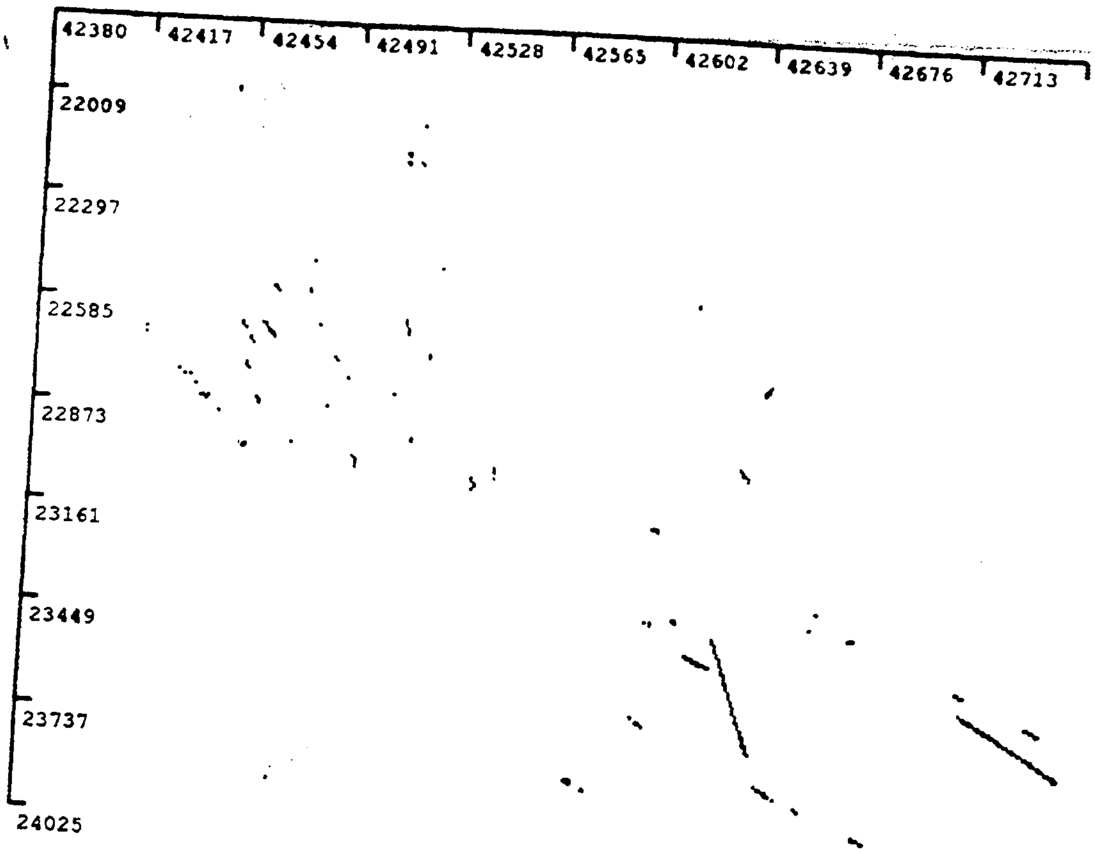

Figure A-136. Input data from data set 44.

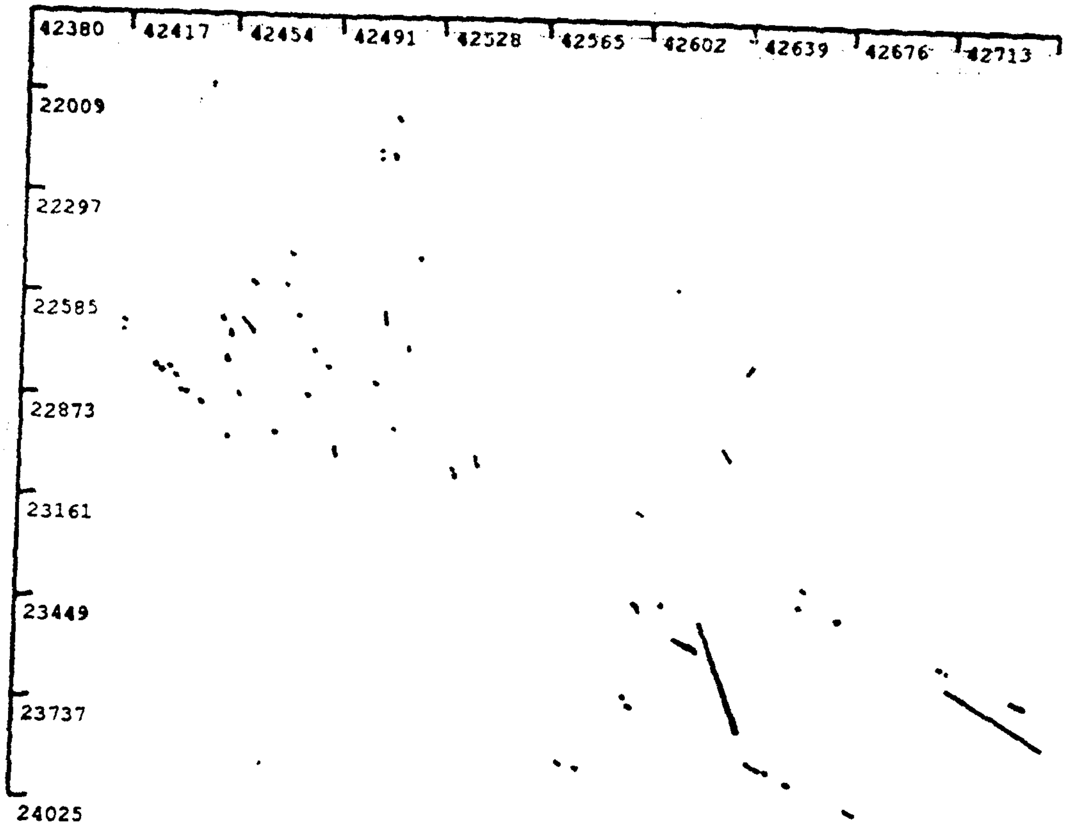

Figure A-137. Initial clusters from data set 44.

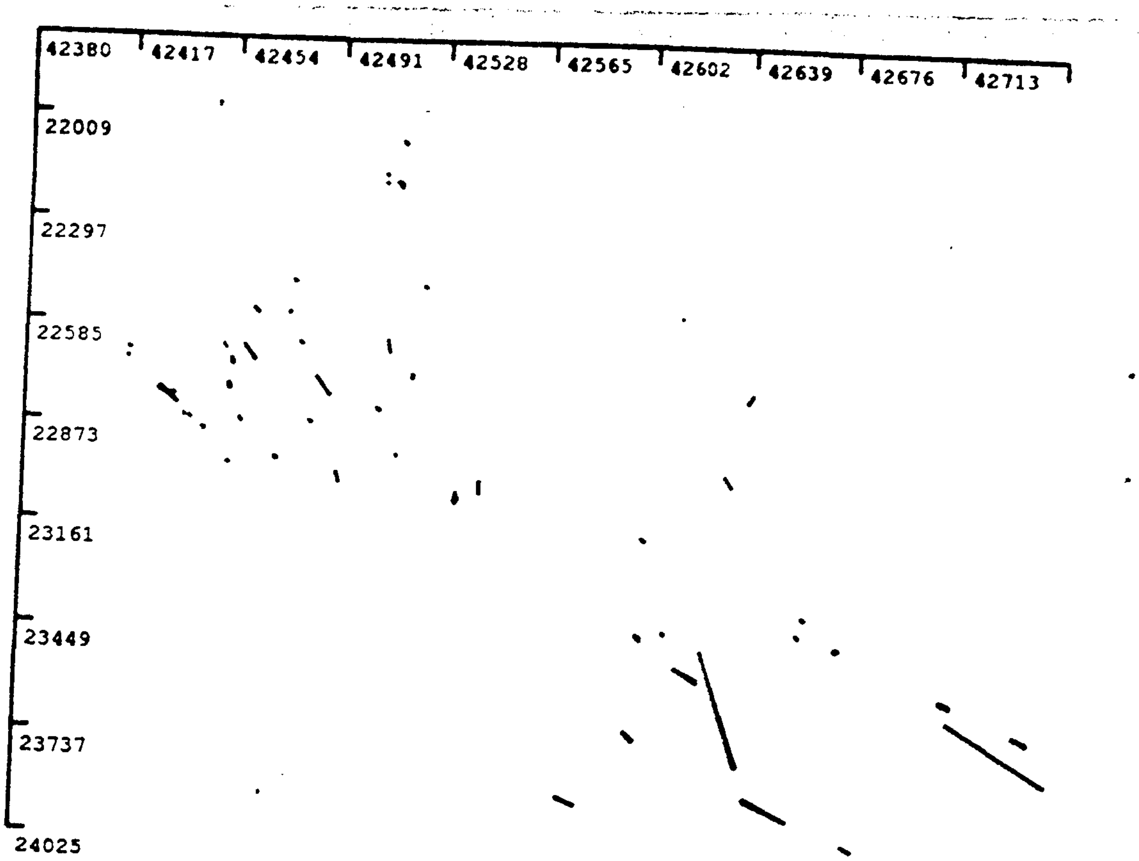

Figure A-138. Linear and online clusters from data set 44.

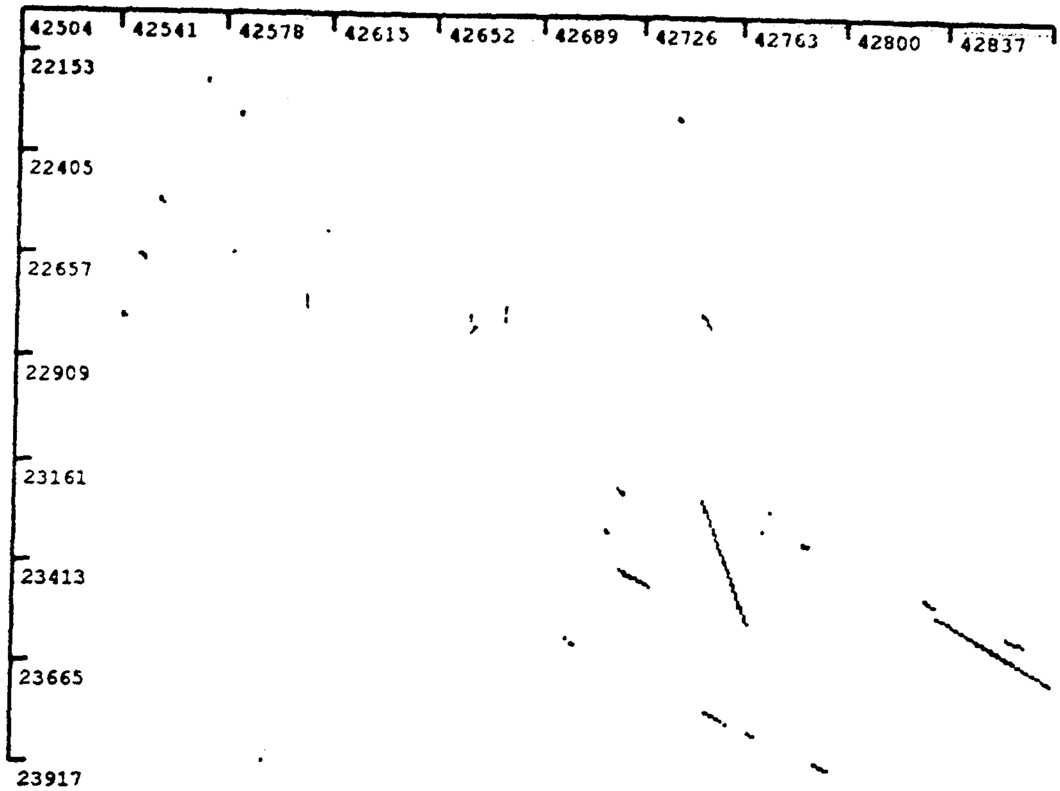

Figure A-139. Input data from data set 45.

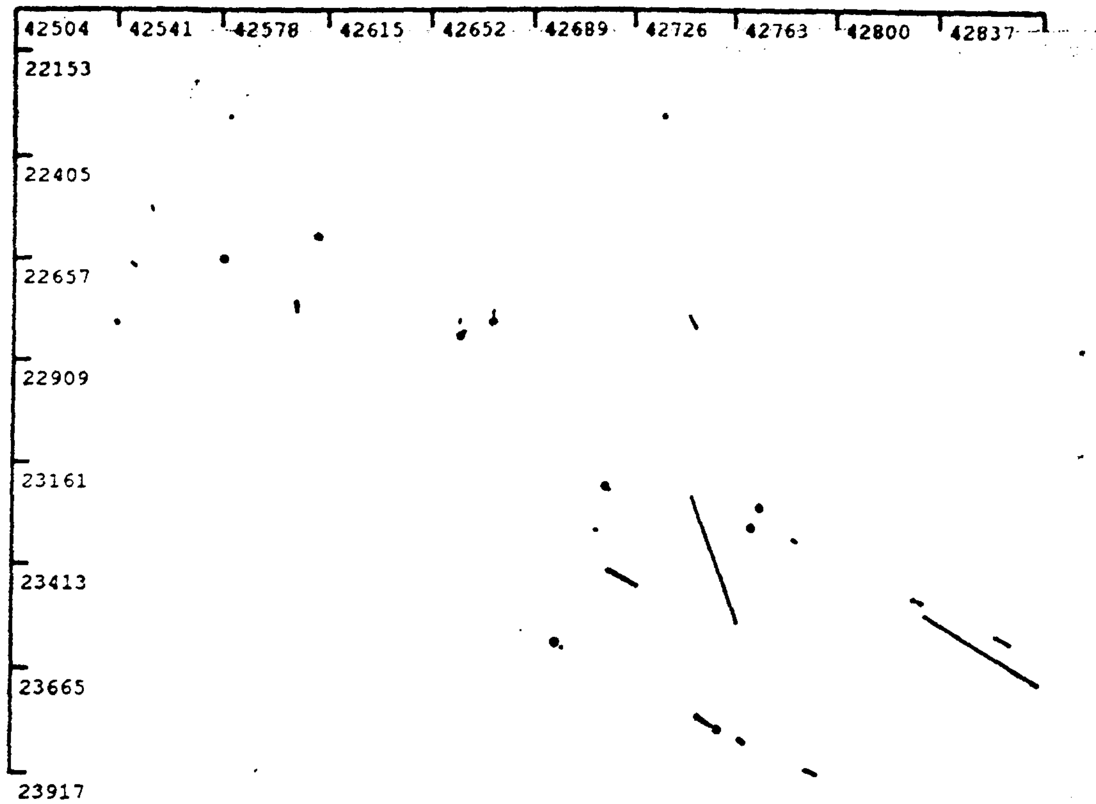

Figure A-140. Initial clusters from data set 45.

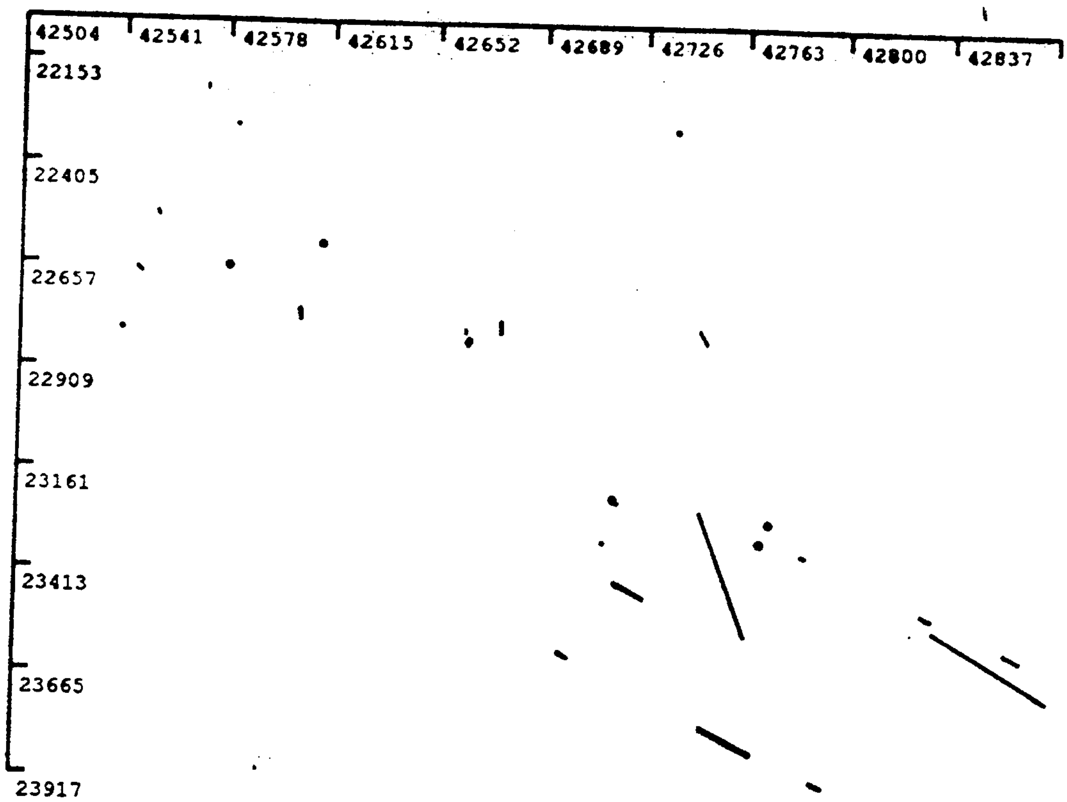

Figure A-141. Linear and online clusters from data set 45.

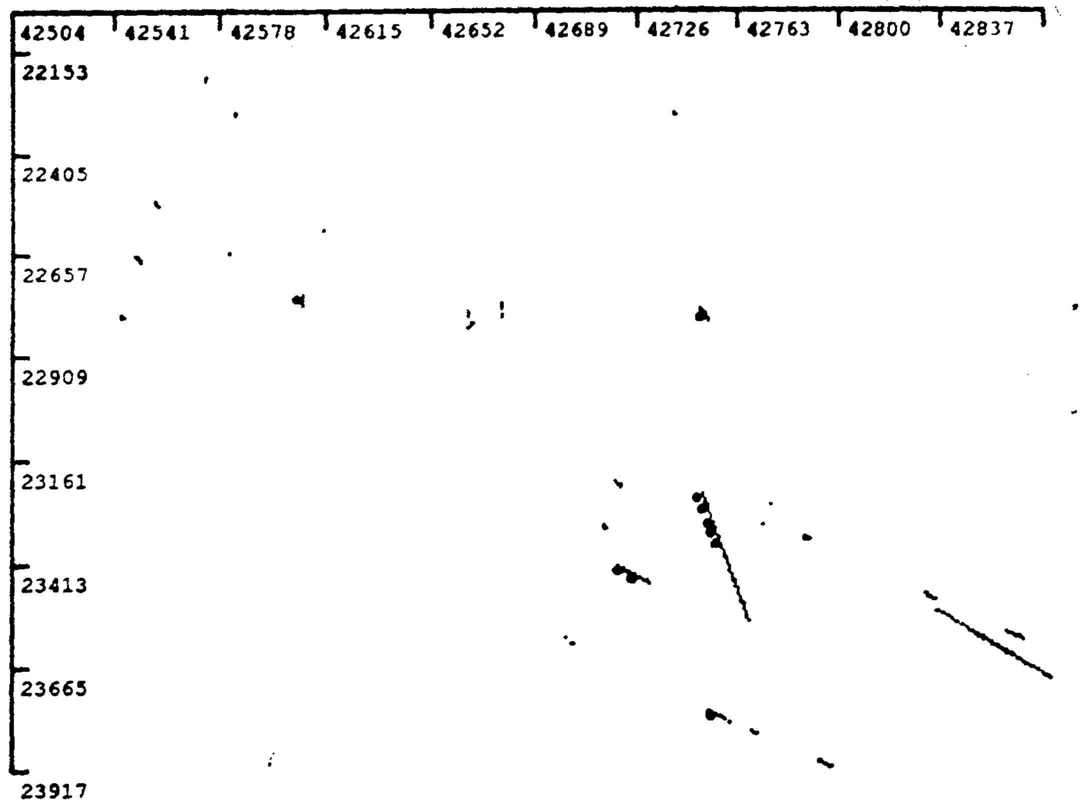

Figure A-142. Circular clustering from data set 45.

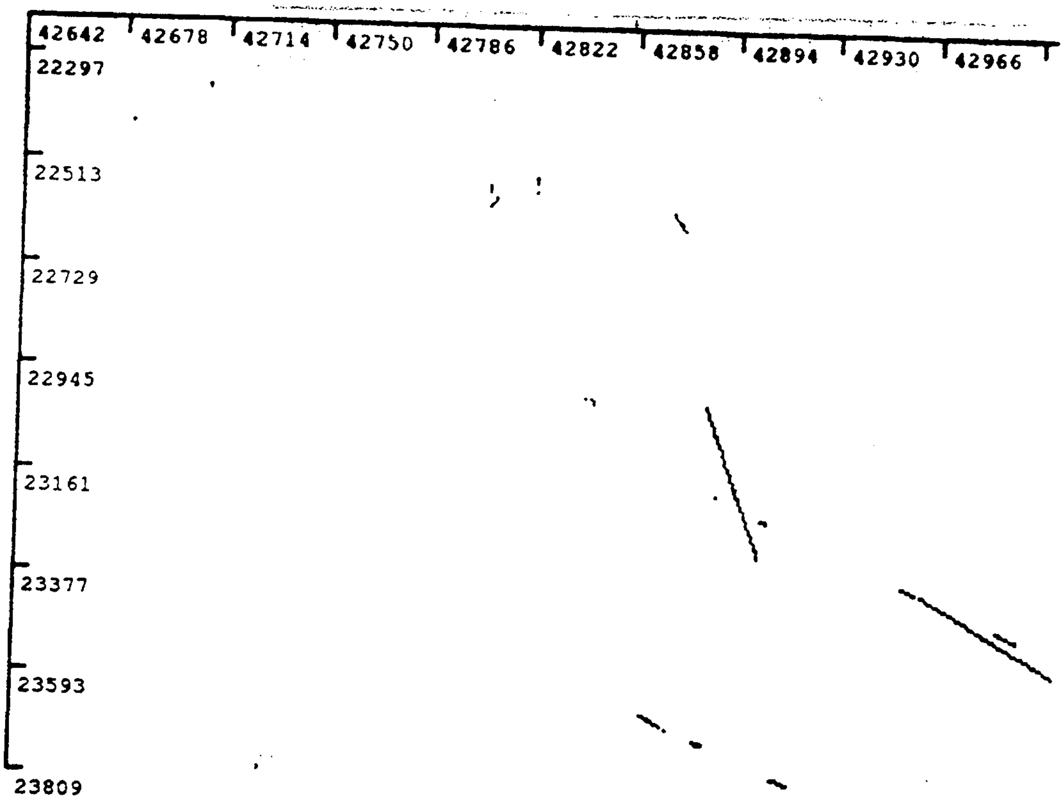

Figure A-143. Input data from data set 46.

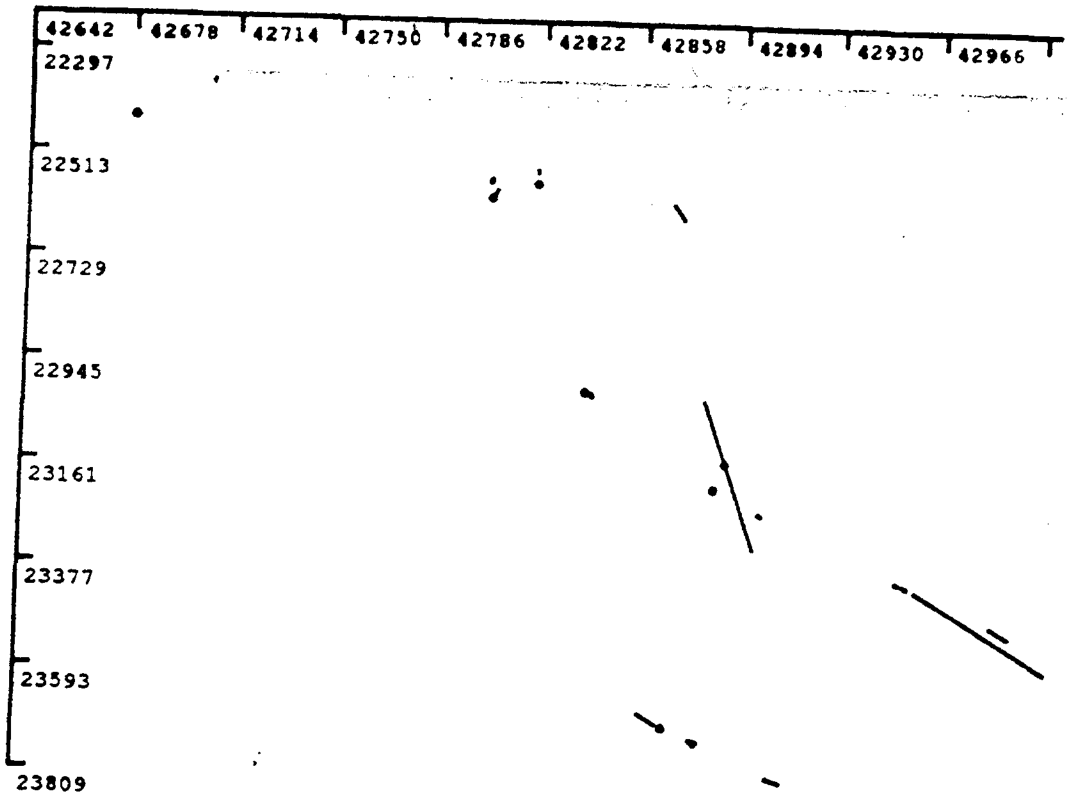

Figure A-144. Initial clusters from data set 46.

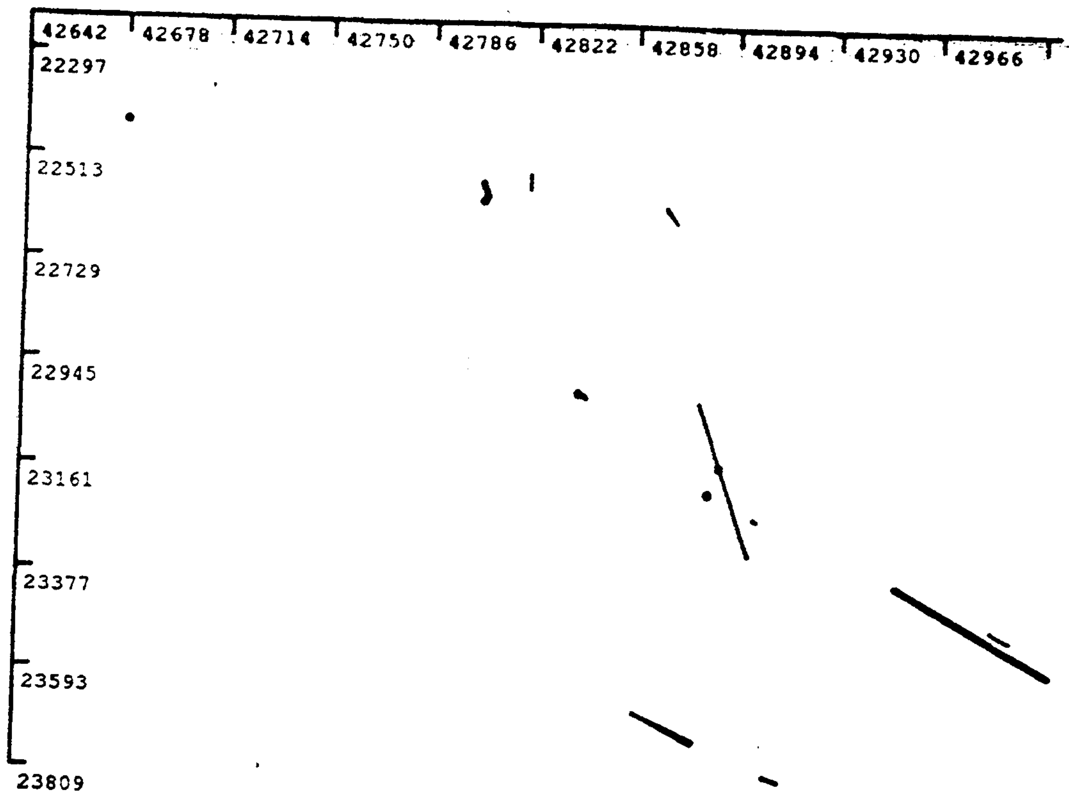

Figure A-145. Linear and online clusters from data set 46.

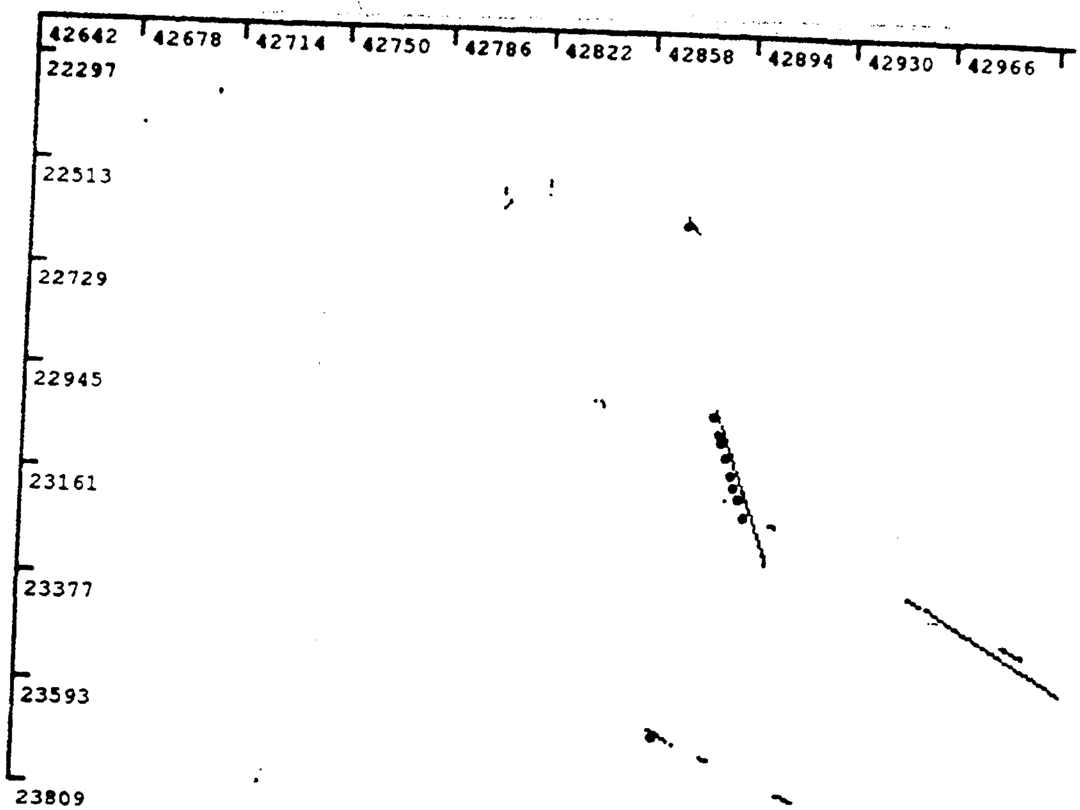

Figure A-146. Circular clustering from data set 46.

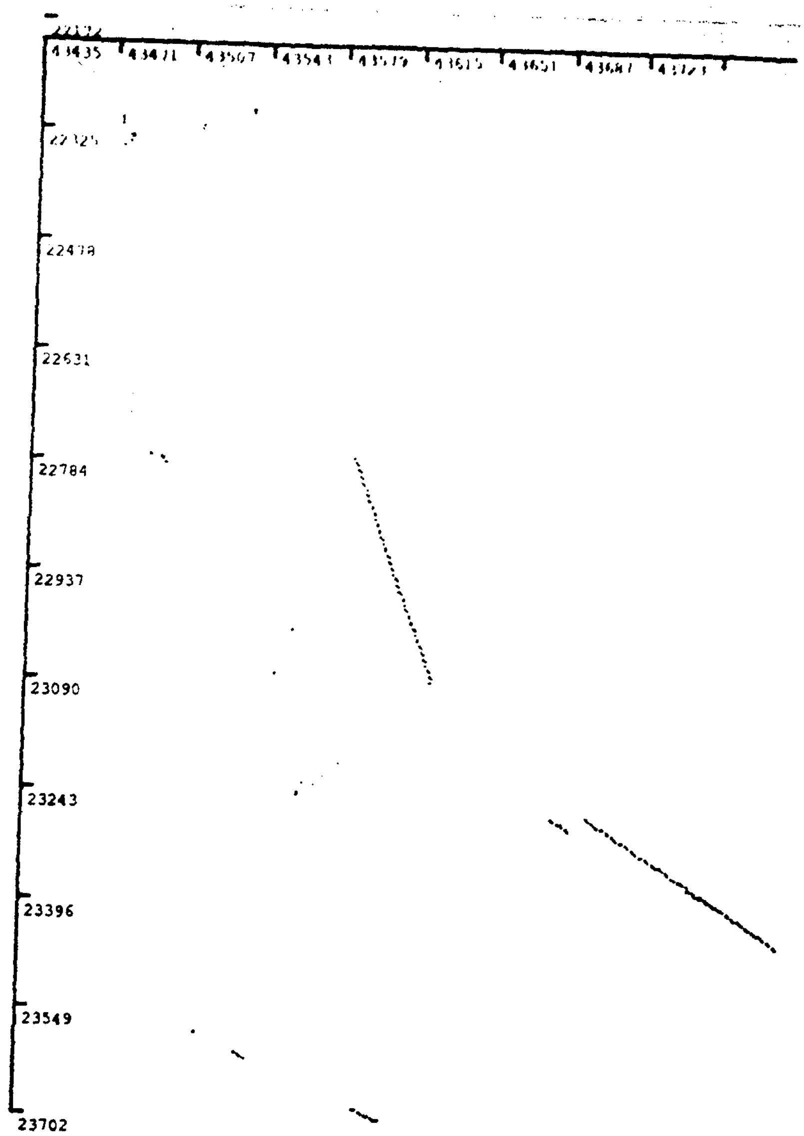

Figure A-147. Input data from data set 47.

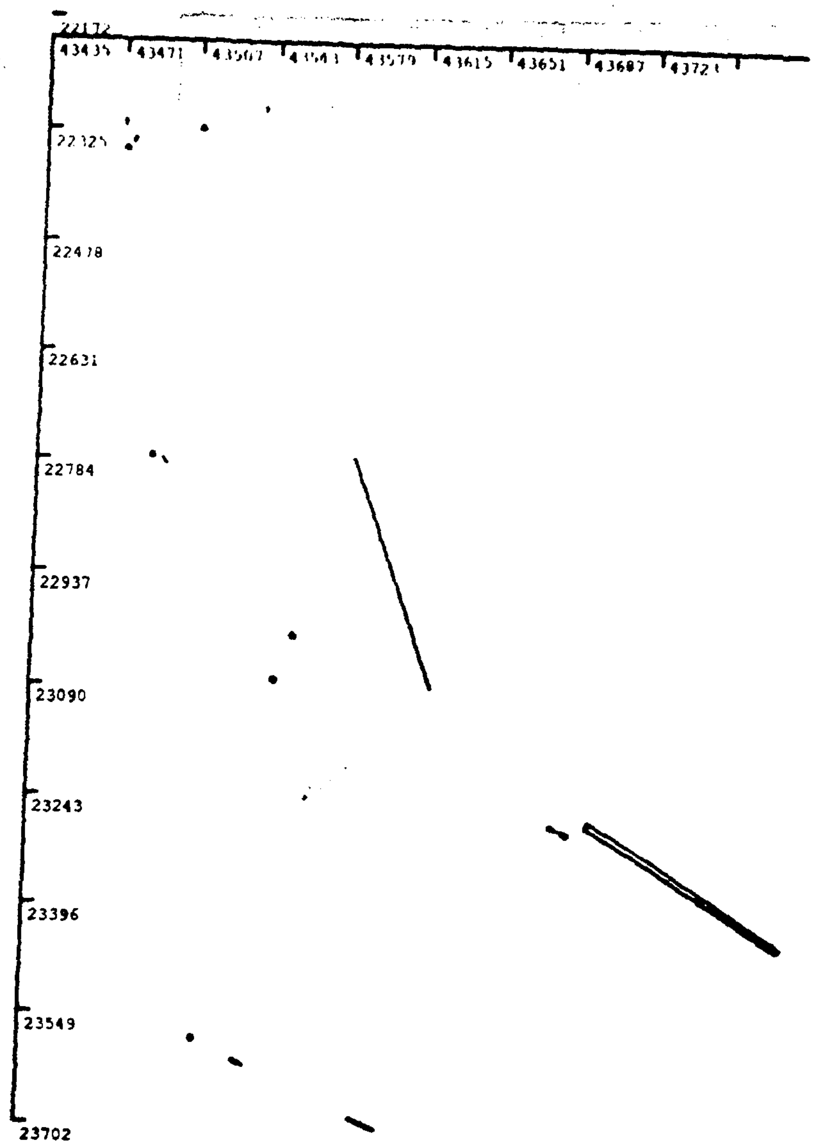

Figure A-148. Initial clusters from data set 47.

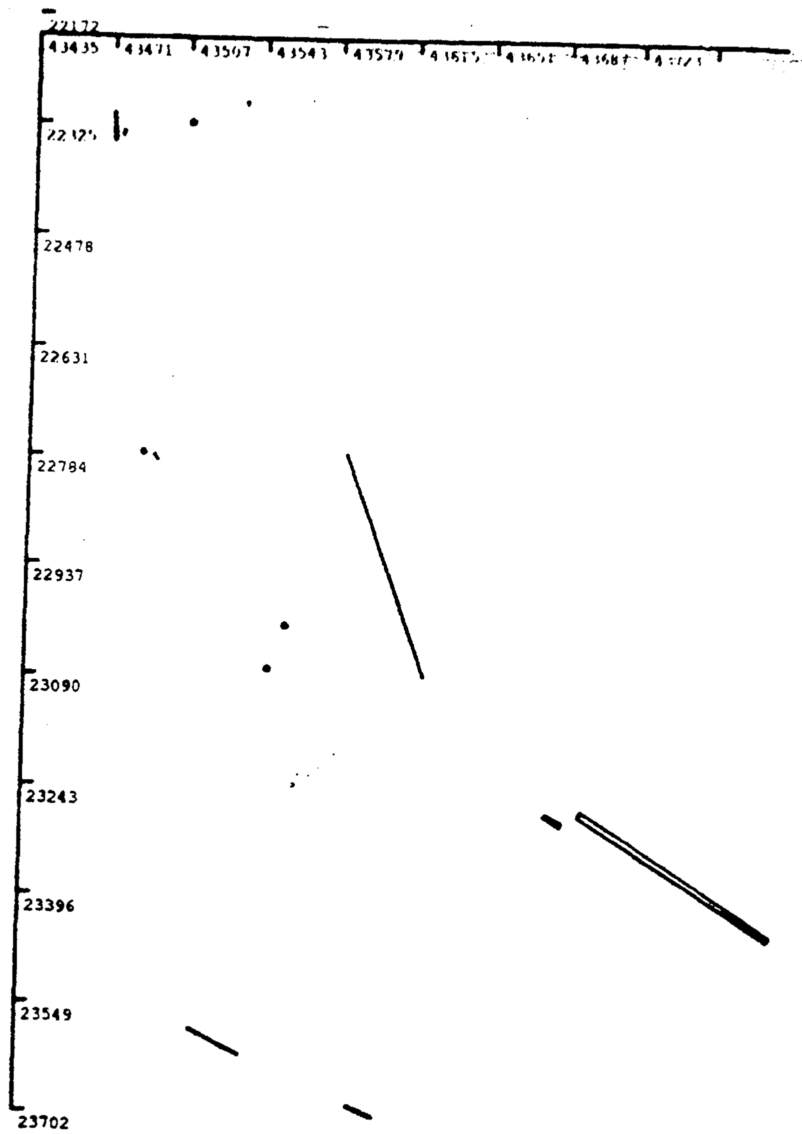

Figure A-149. Linear and online clusters from data set 47.

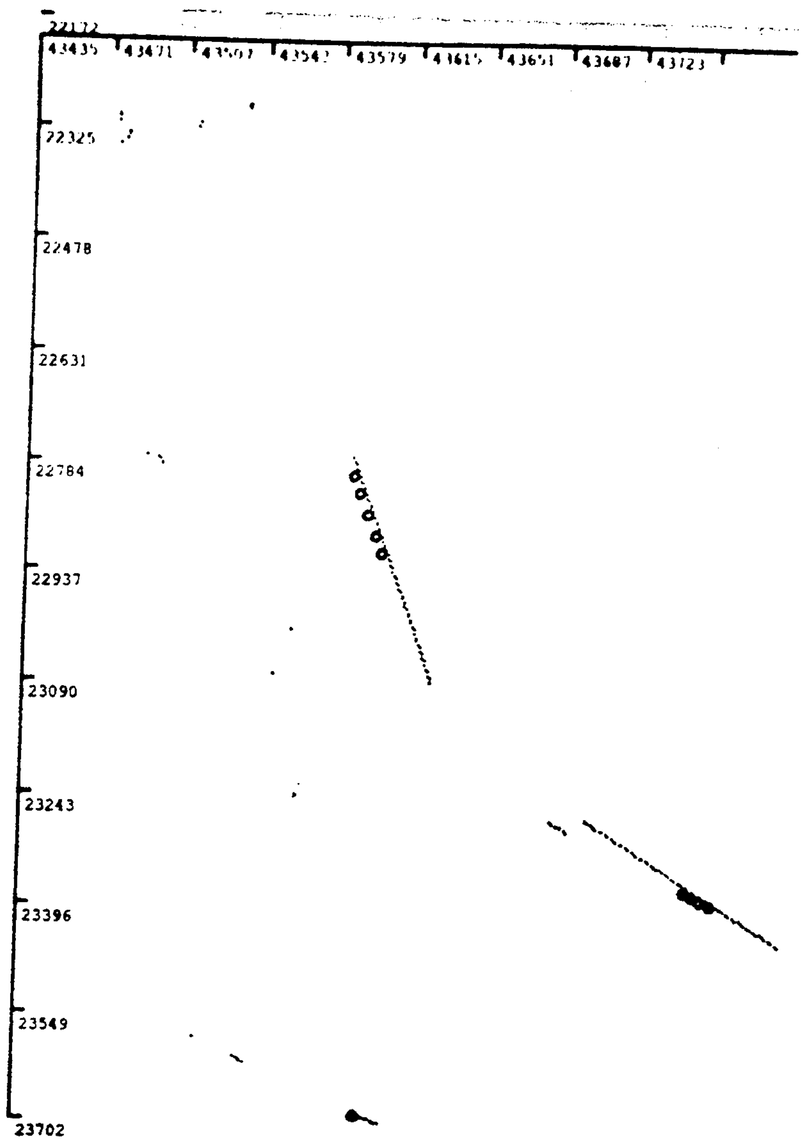

Figure A-150. Circular clustering from data set 47.

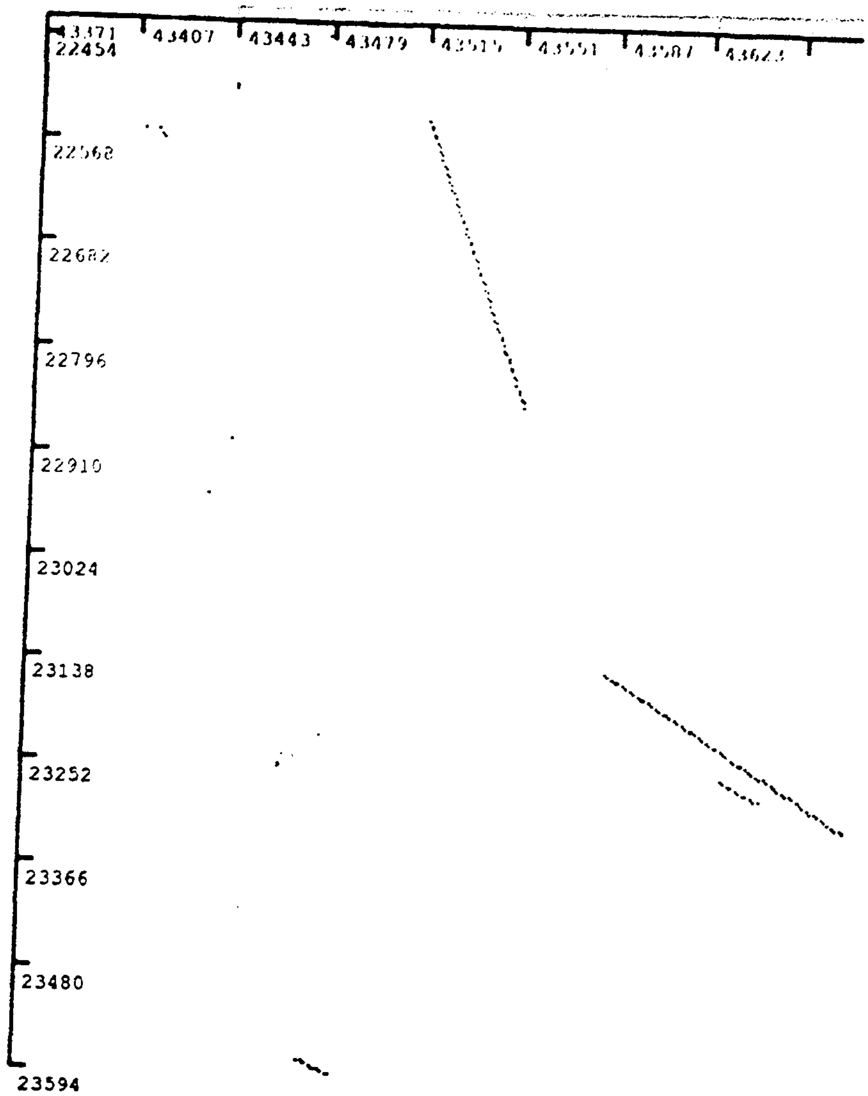

Figure A-151. Input data from data set 48.

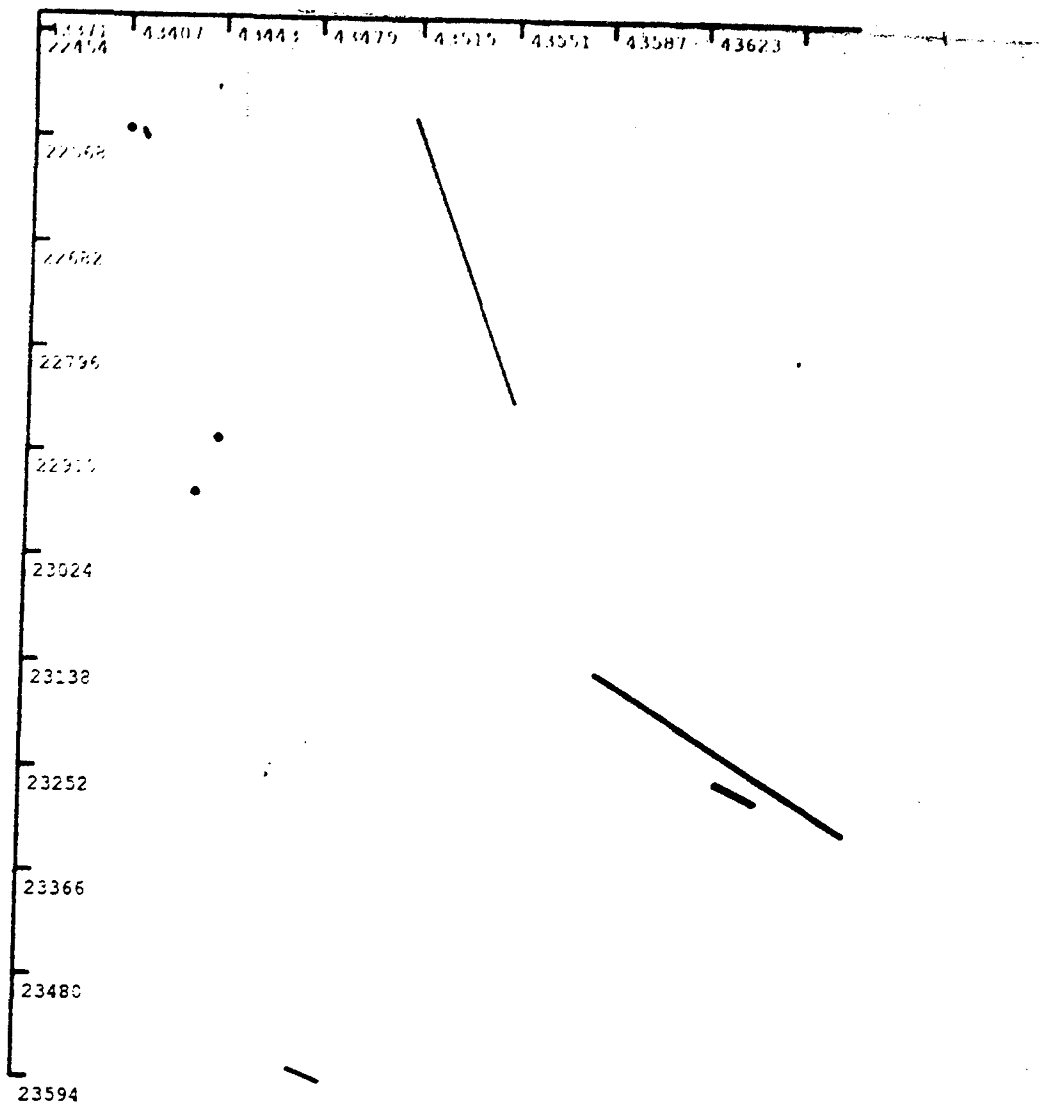

Figure A-152. Initial clusters from data set 48.

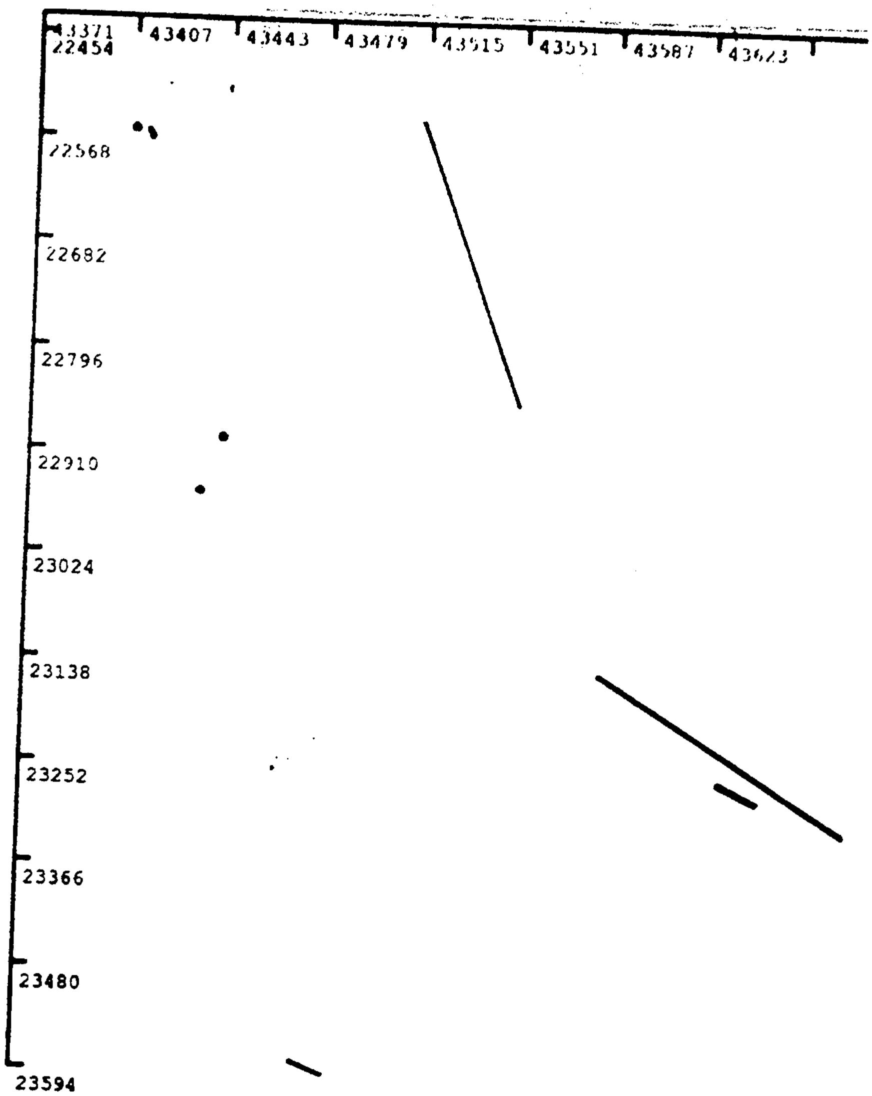

Figure A-153. Linear and online clusters from data set 48.

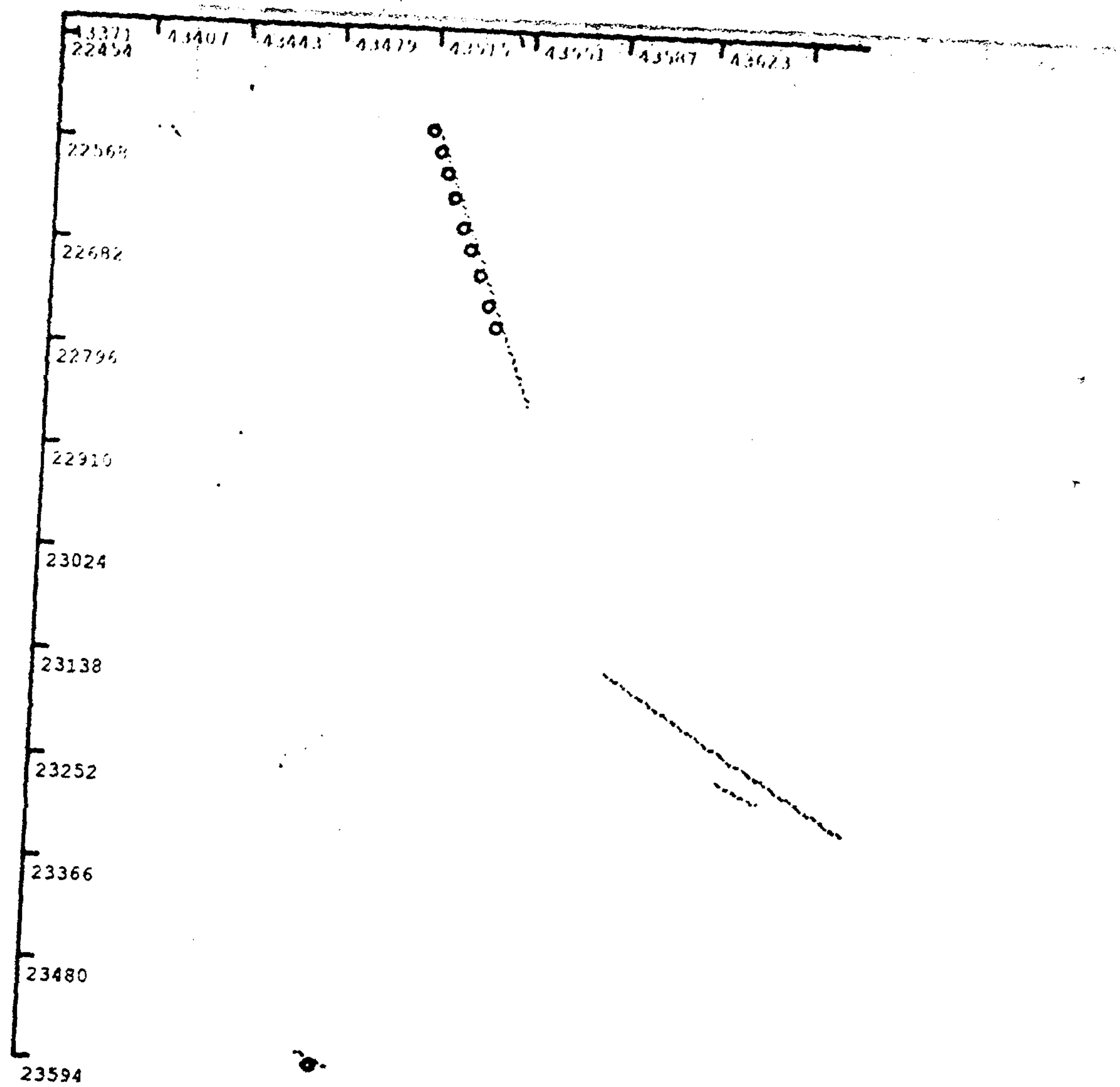

Figure A-154. Circular clustering from data set 48.

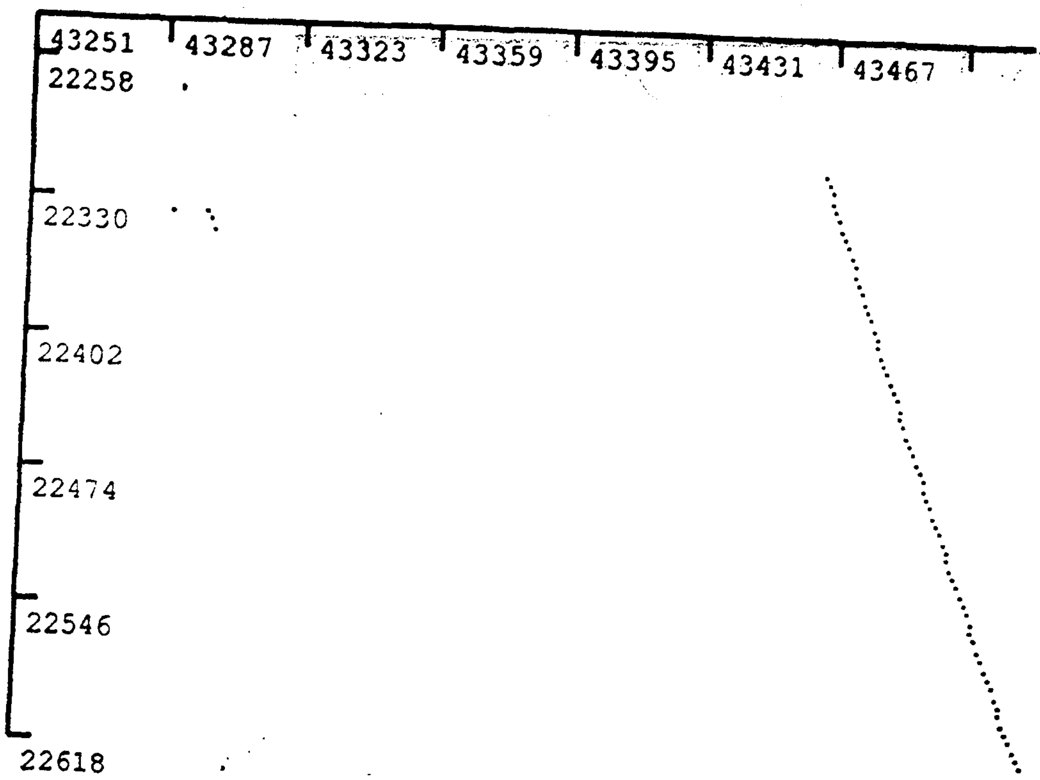

Figure A-155. Input data from data set 49.

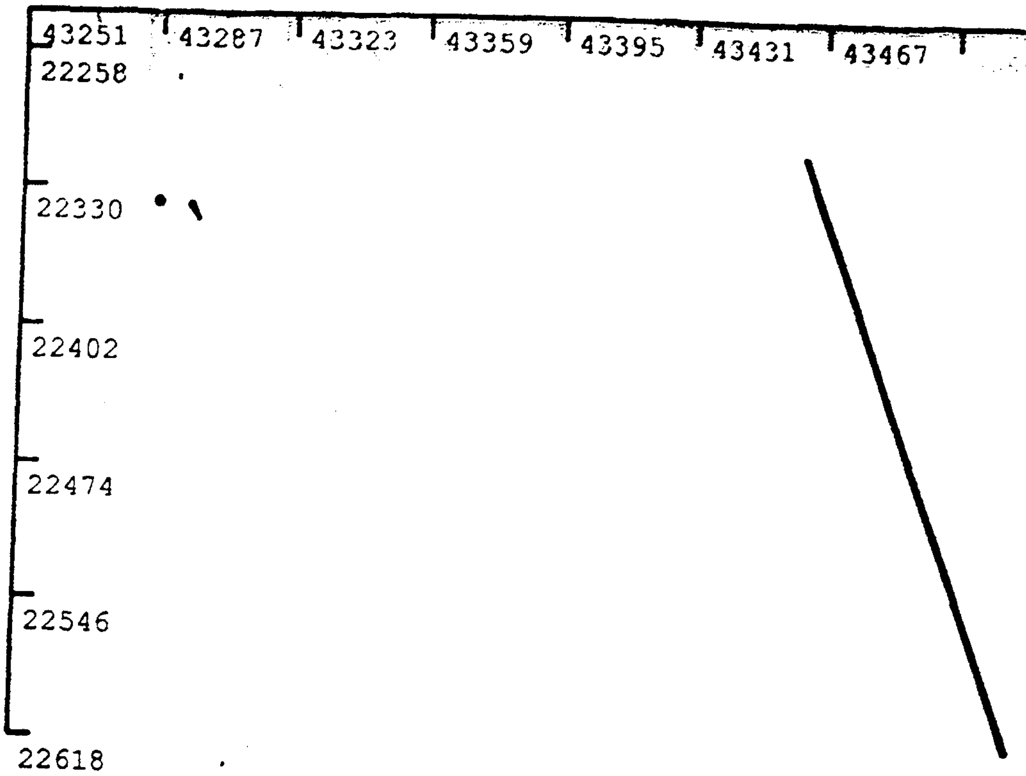

Figure A-156. Initial clusters from data set 49.

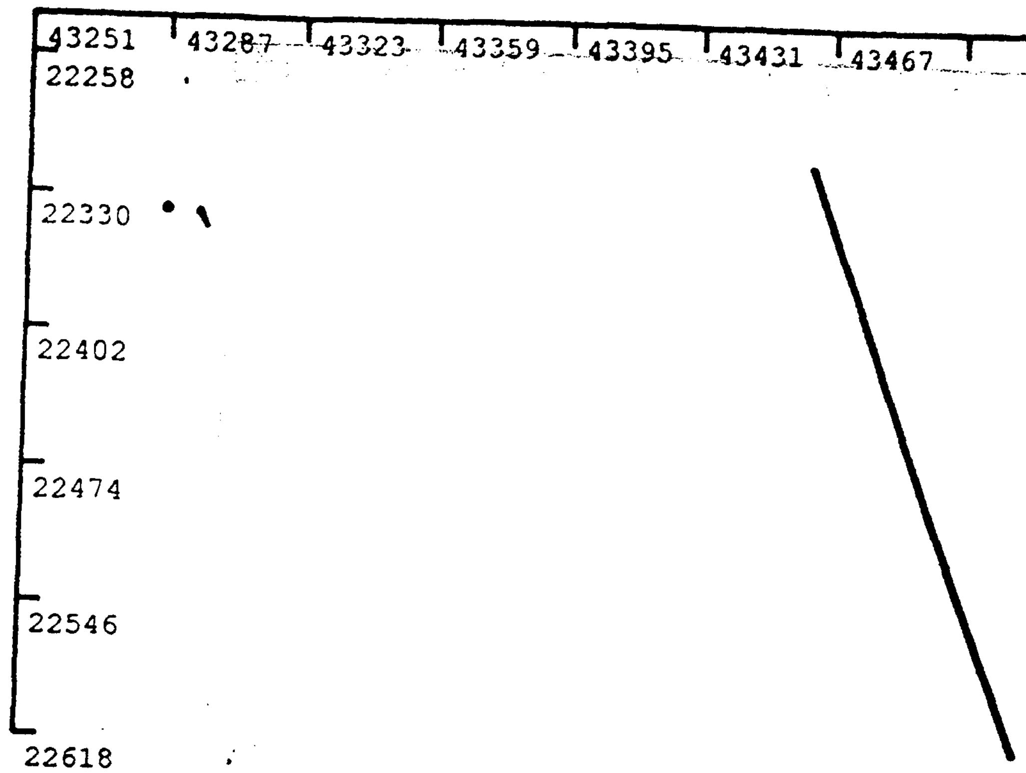

Figure A-157. Linear and online clusters from data set 49.

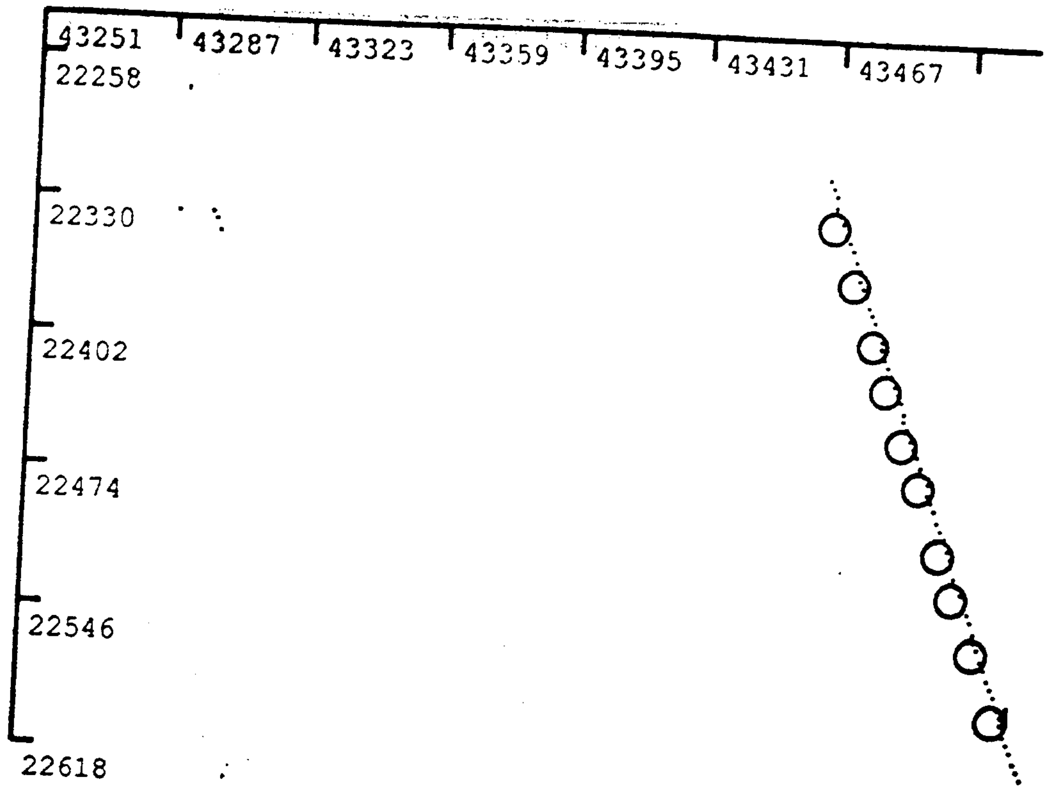

Figure A-158. Circular clustering from data set 49.

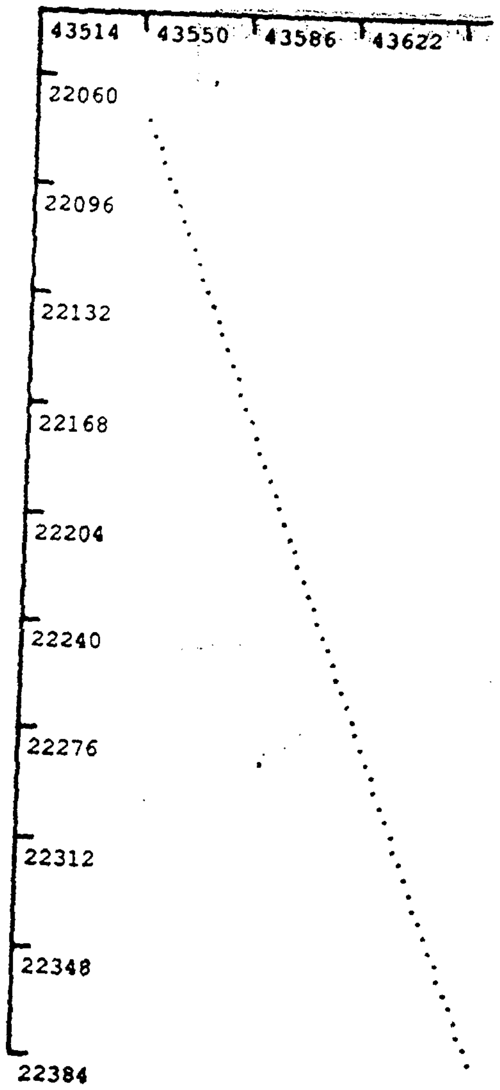

Figure A-159. Input data from data set 50.

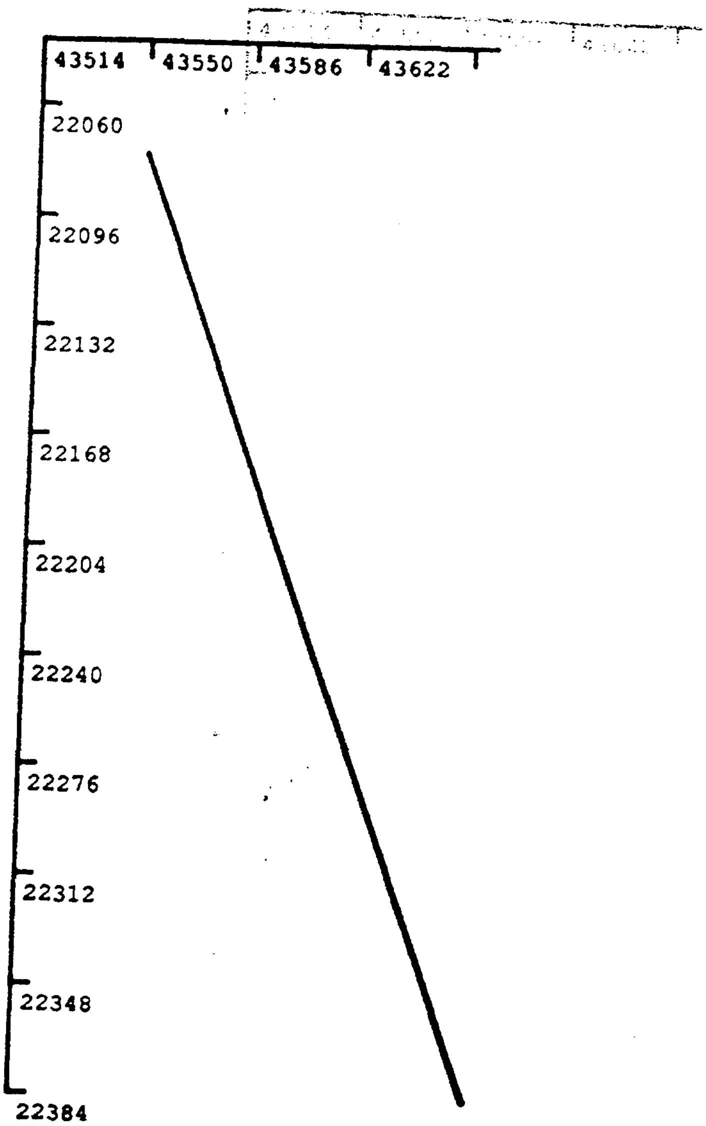

Figure A-160. Initial clusters from data set 50.

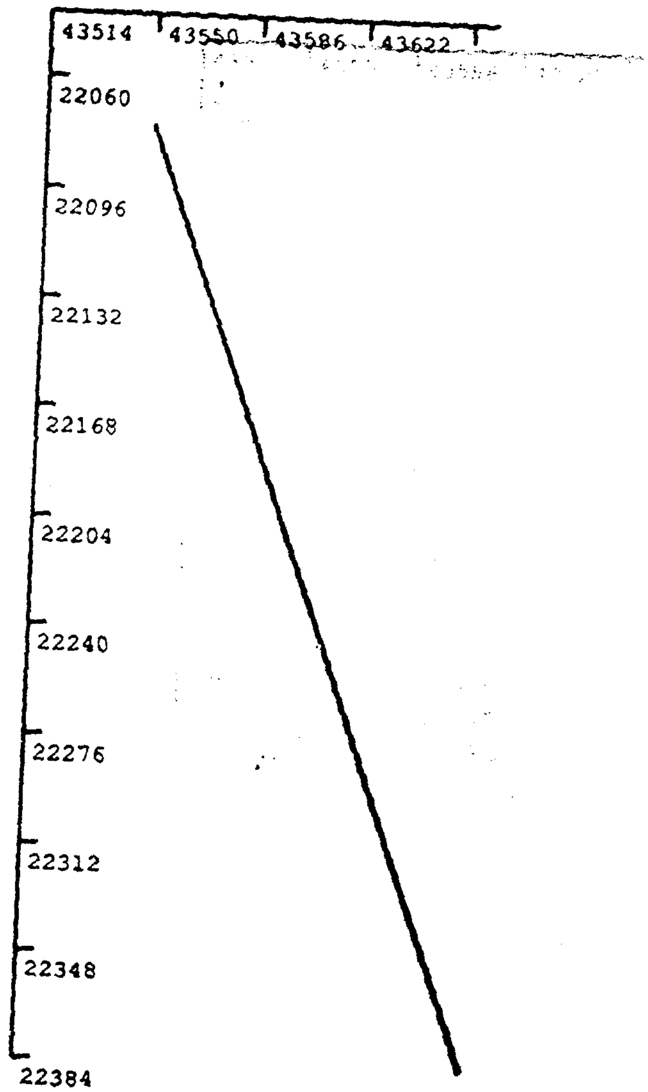

Figure A-161. Linear and online clusters from data set 50.

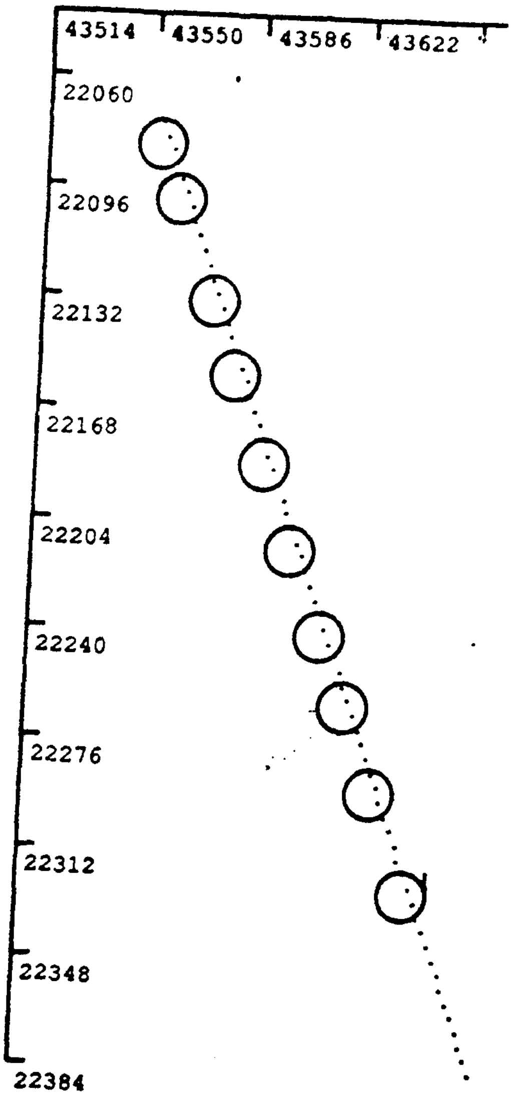

Figure A-162. Circular clustering from data set 50.

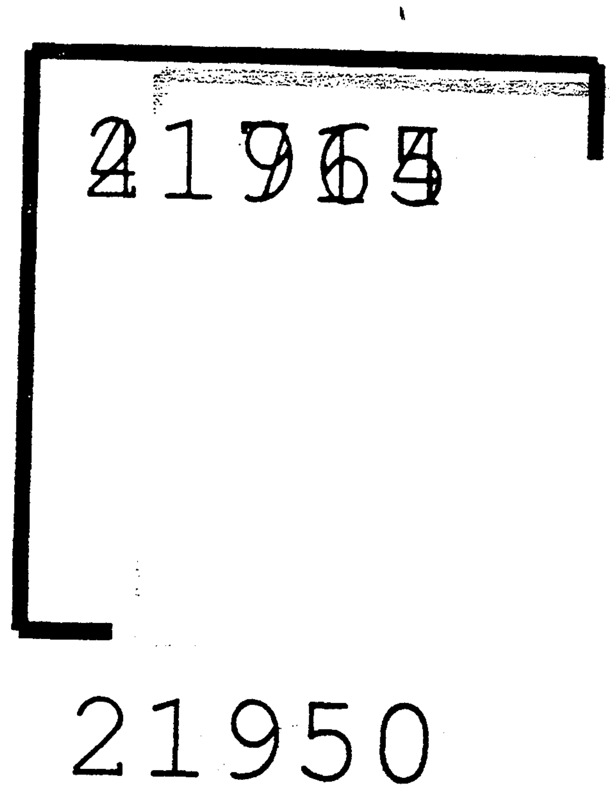

Figure A-163. Input data from data set 51.

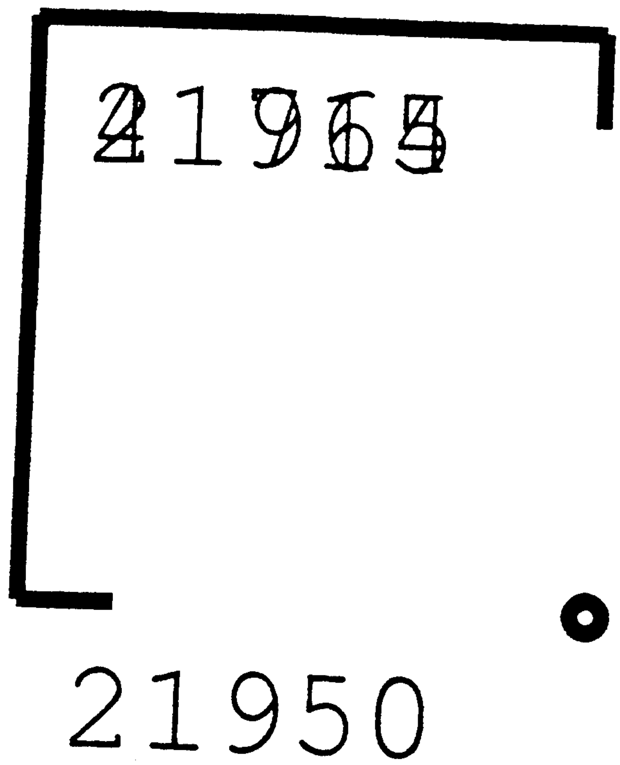

Figure A-164. Initial clusters from data set 51.

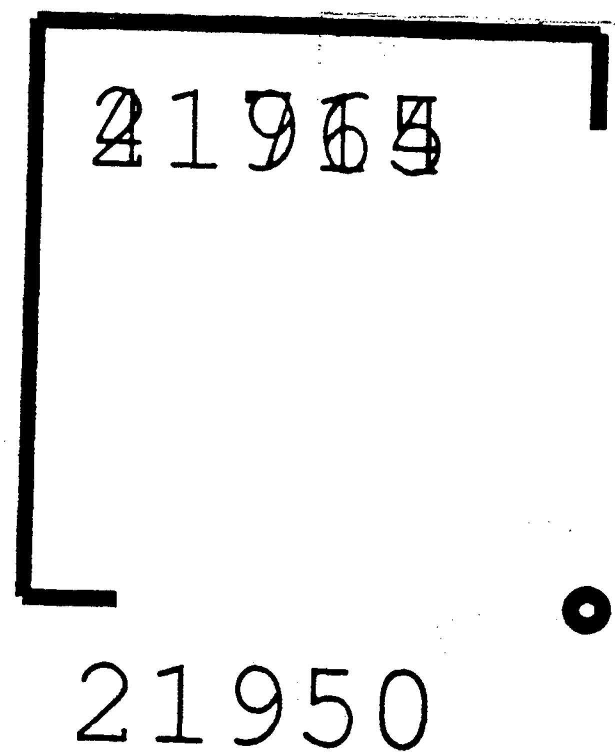

Figure A-165. Linear and online clusters from data set 51.

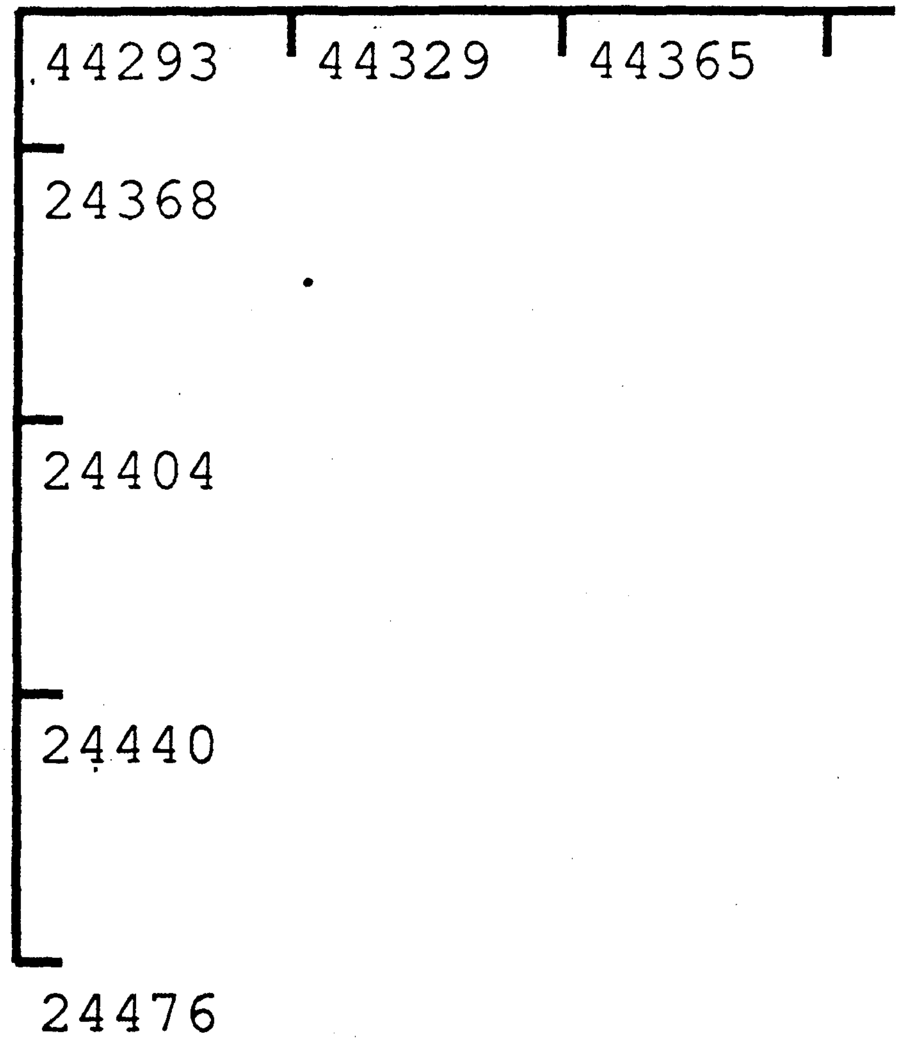

Figure A-166. Input data from data set 52.

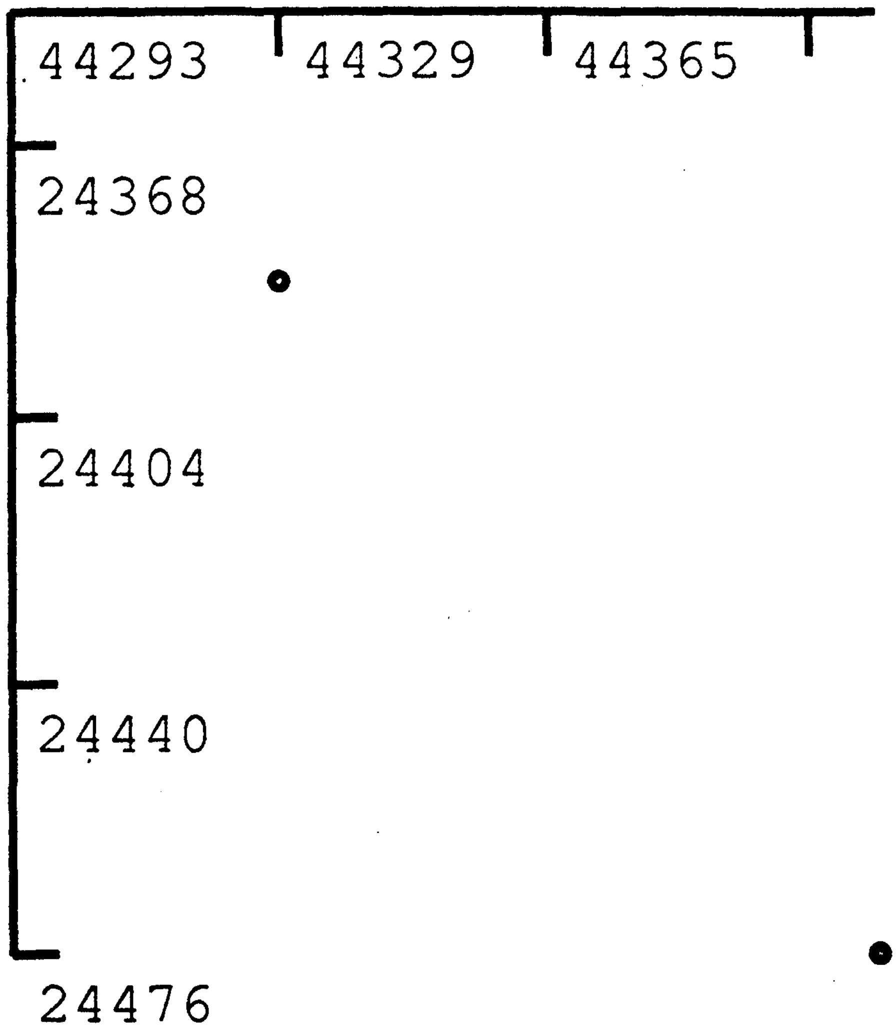

Figure A-167. Initial clusters from data set 52.

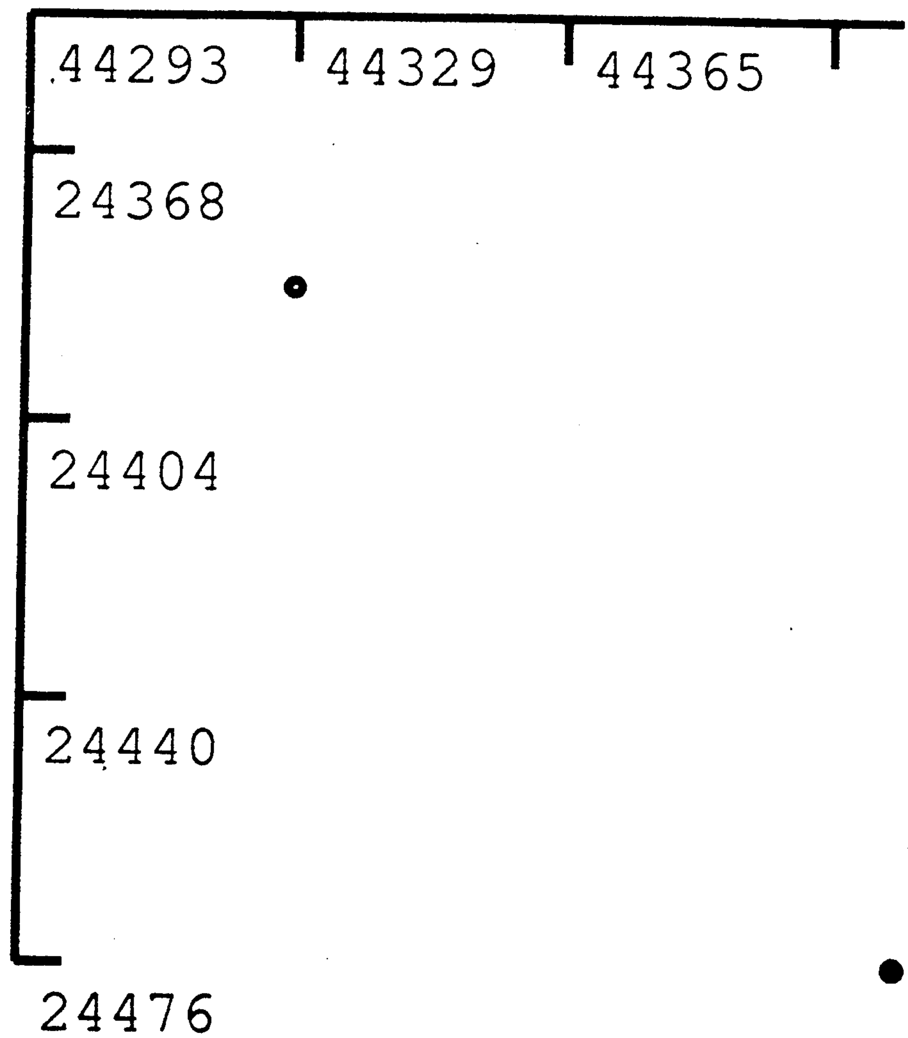

Figure A-168. Linear and online clusters from data set 52.

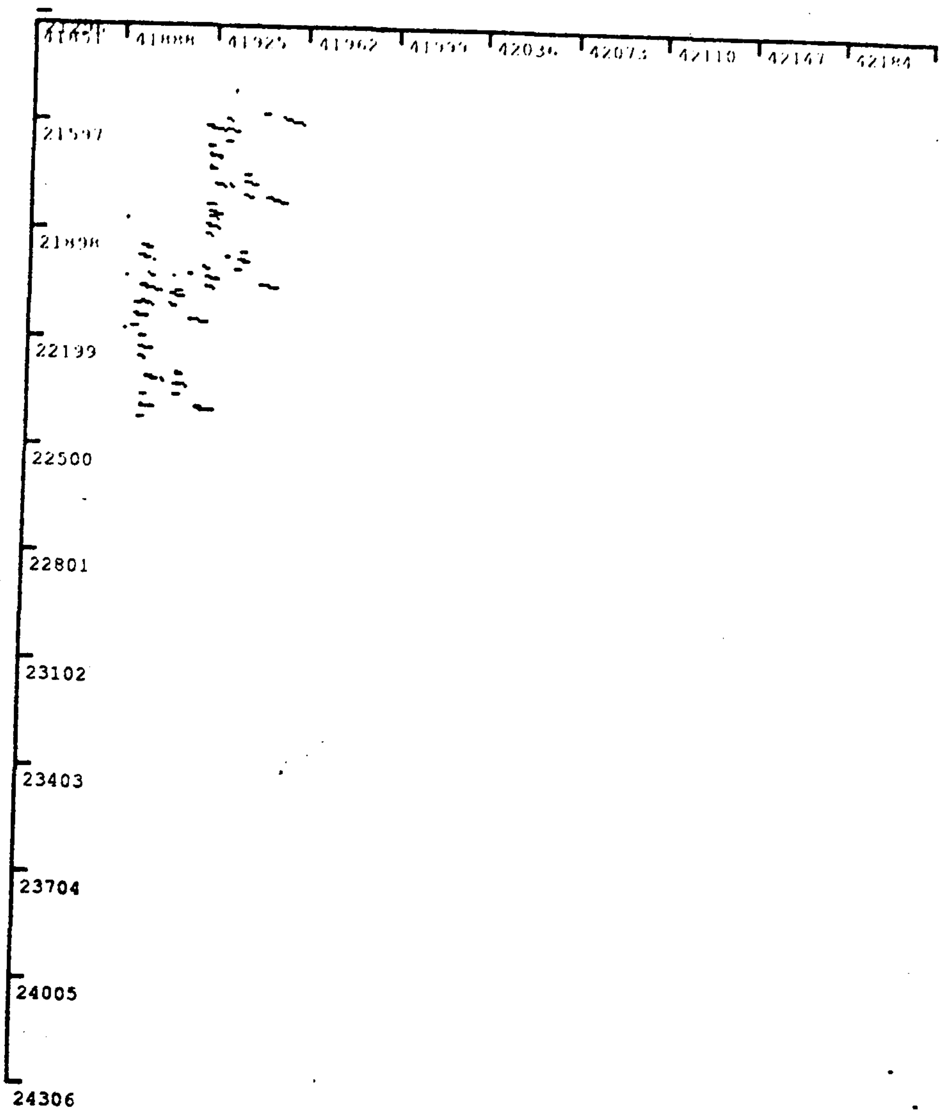

Figure A-169. Input data from data set 53.

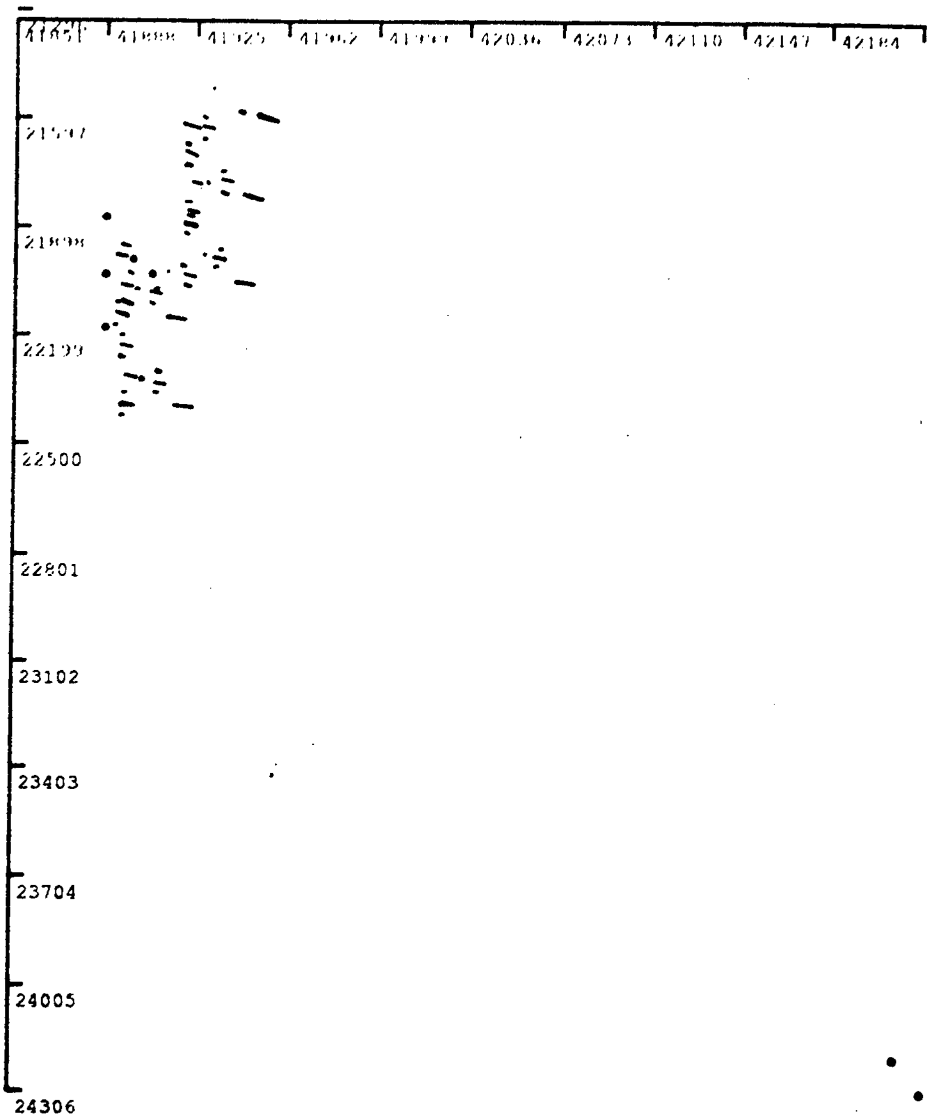

Figure A-170. Initial clusters from data set 53.

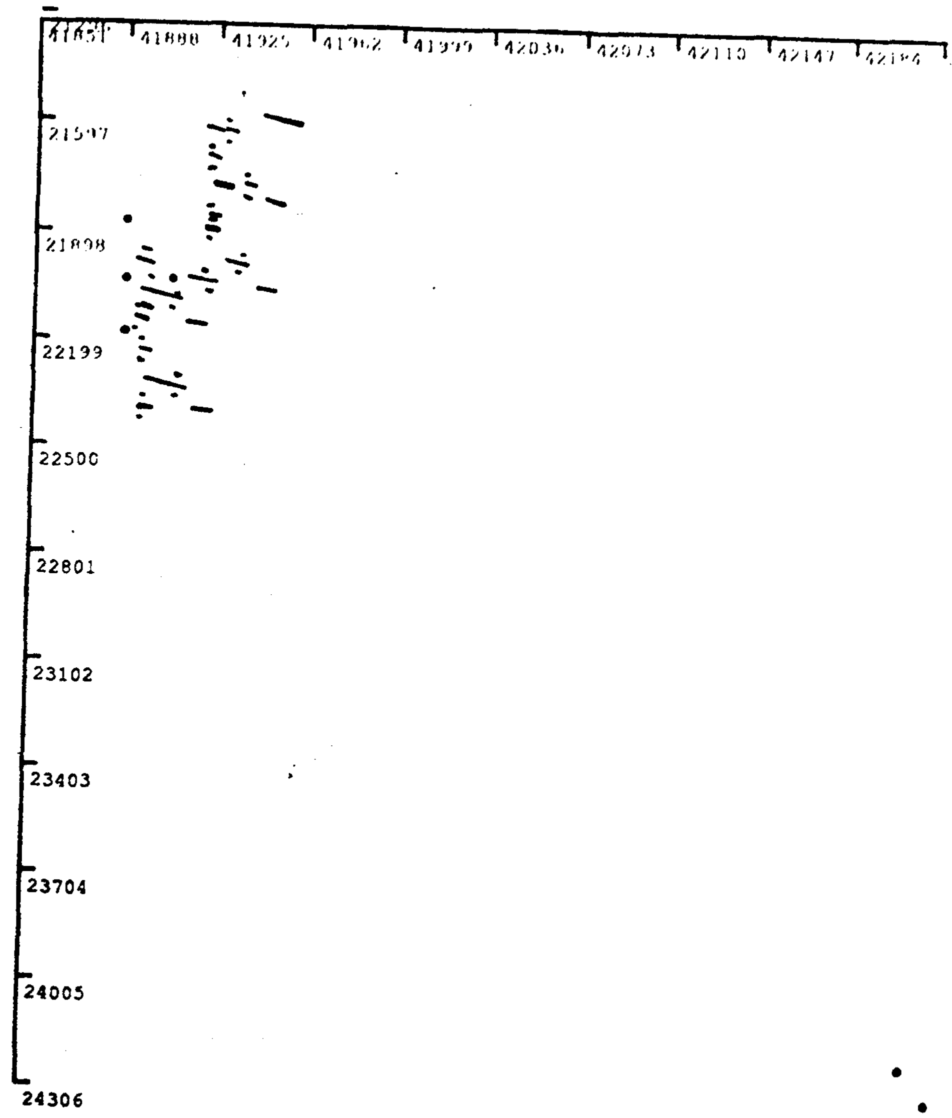

Figure A-171. Linear and online clusters from data set 53.

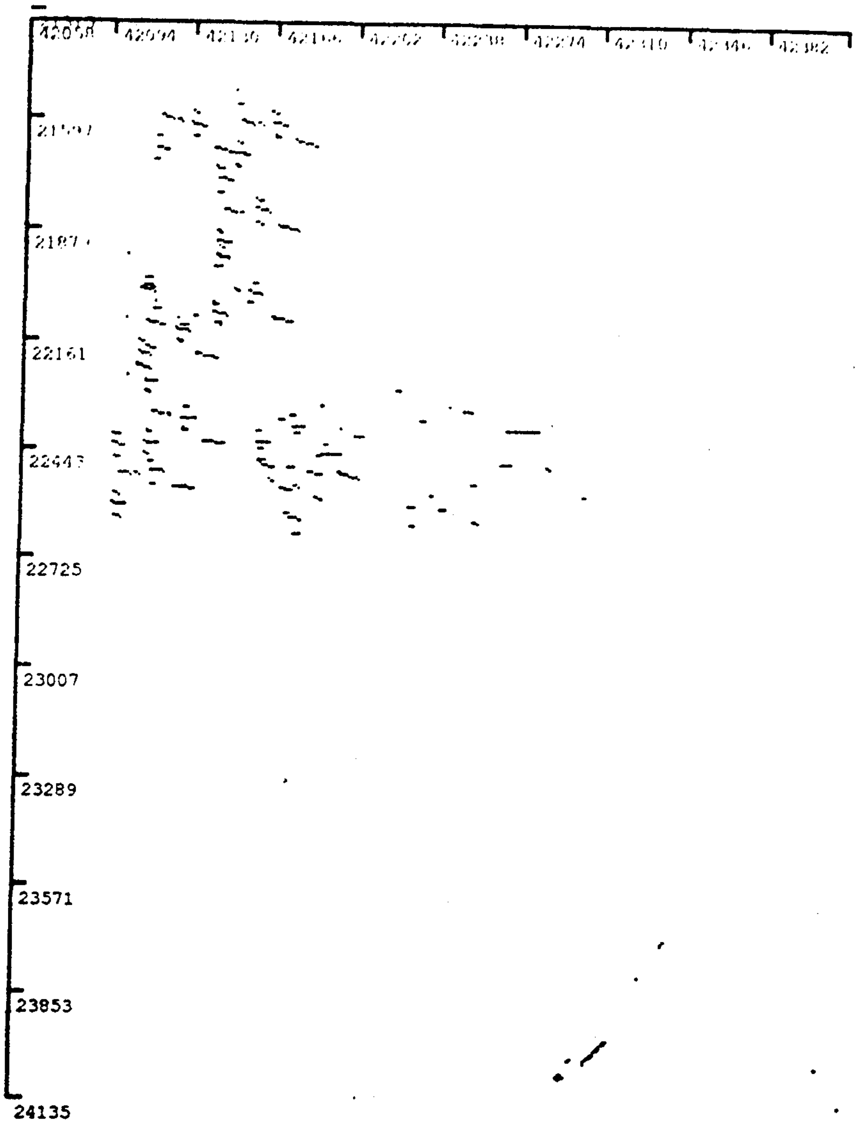

Figure A-172. Input data from data set 54.

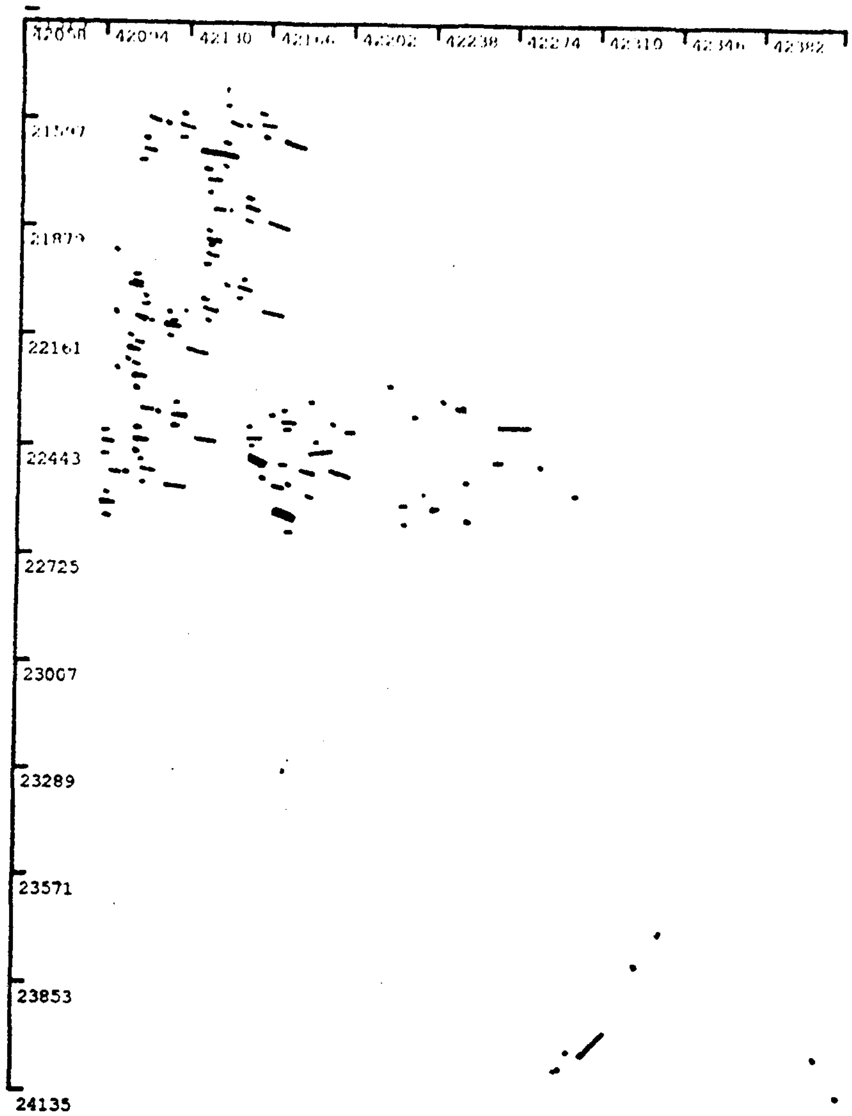

Figure A-173. Initial clusters from data set 54.

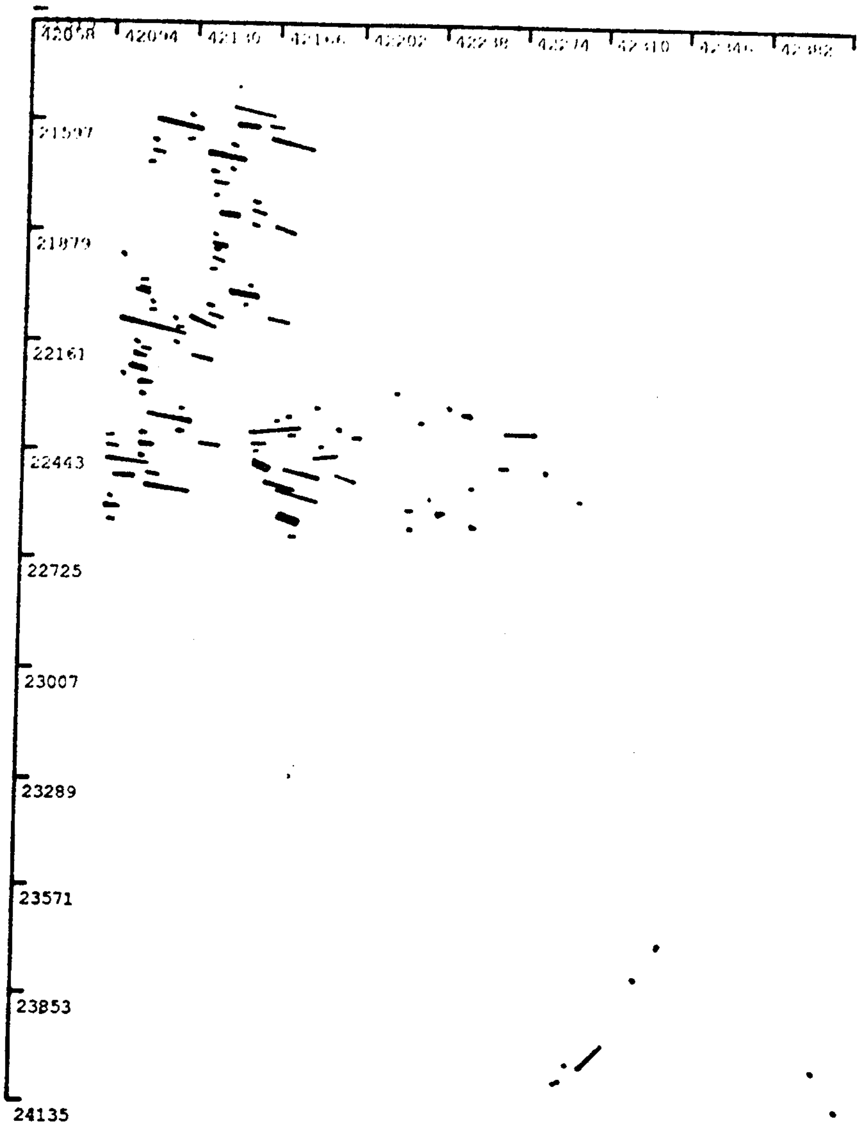

Figure A-174. Linear and online clusters from data set 54.

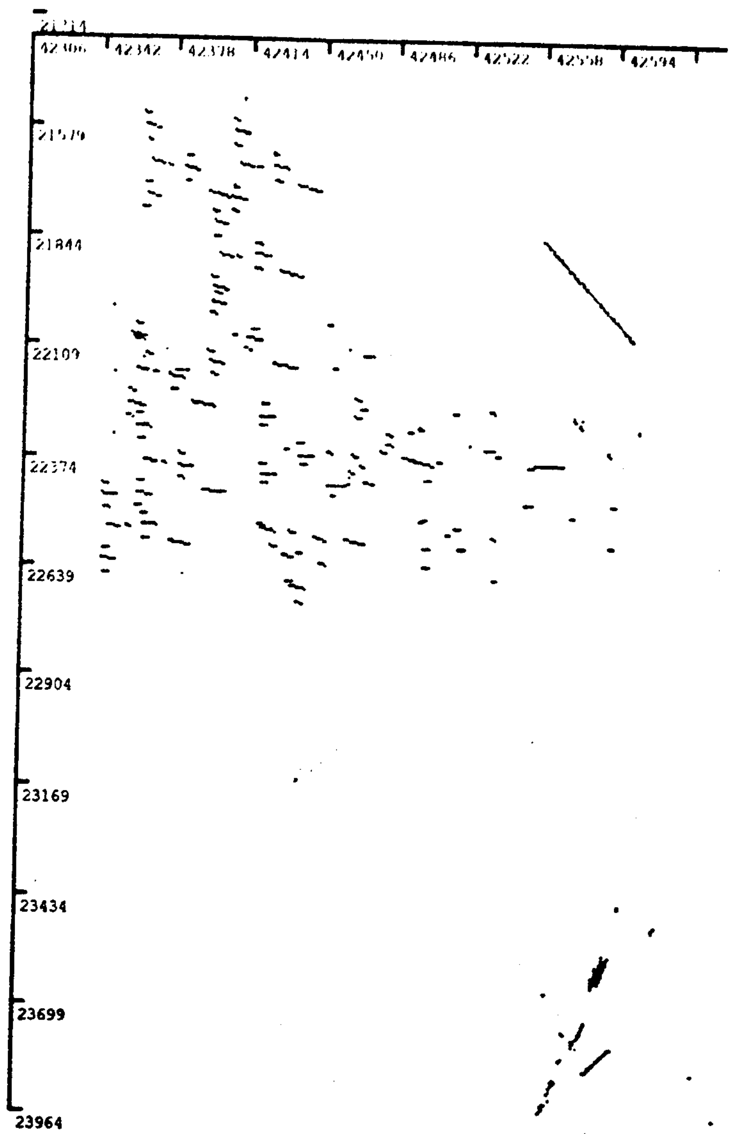

Figure A-175. Input data from data set 55.

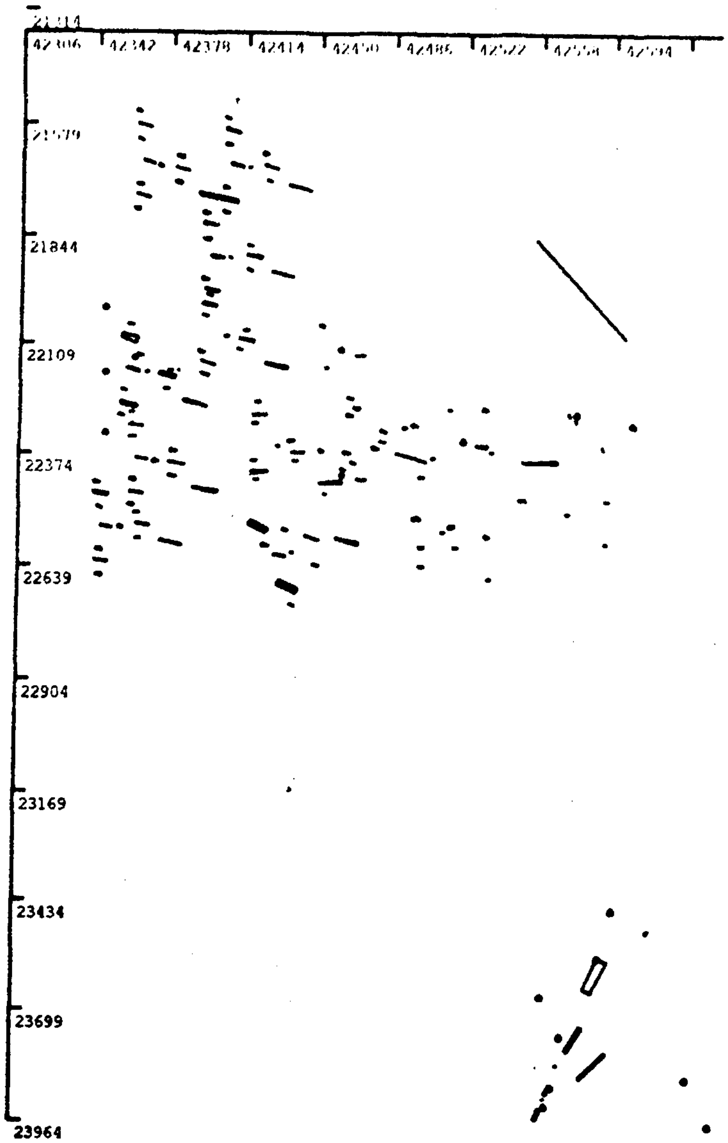

Figure A-176. Initial clusters from data set 55.

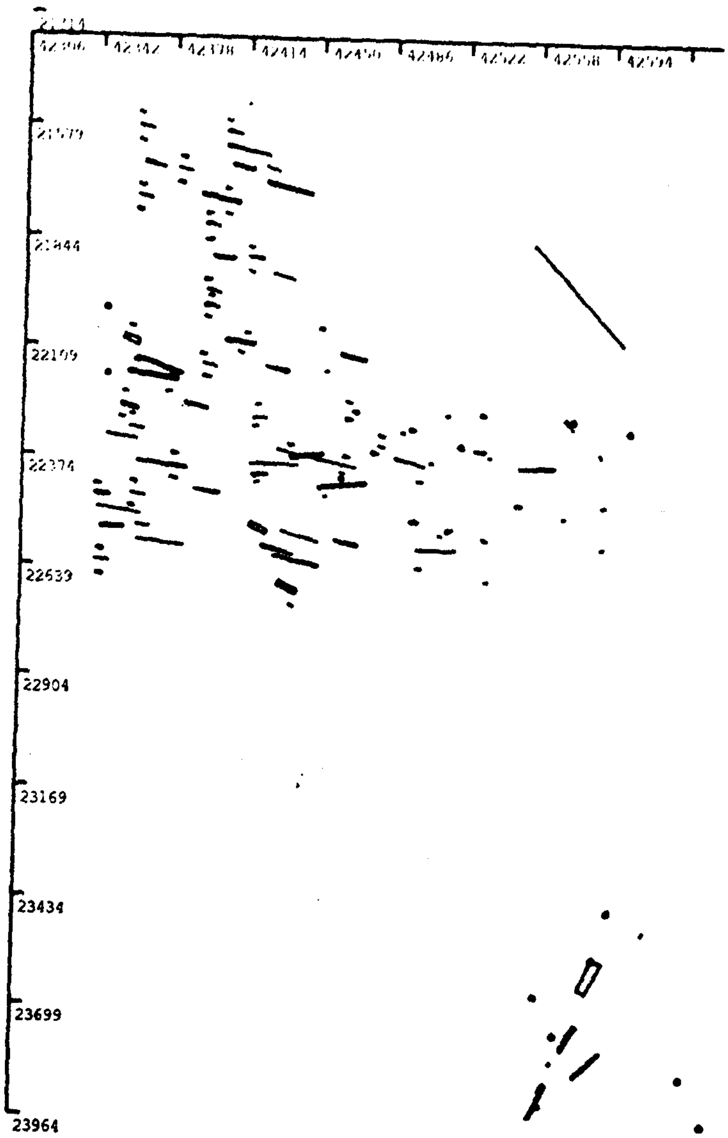

Figure A-177. Linear and online clusters from data set 55.

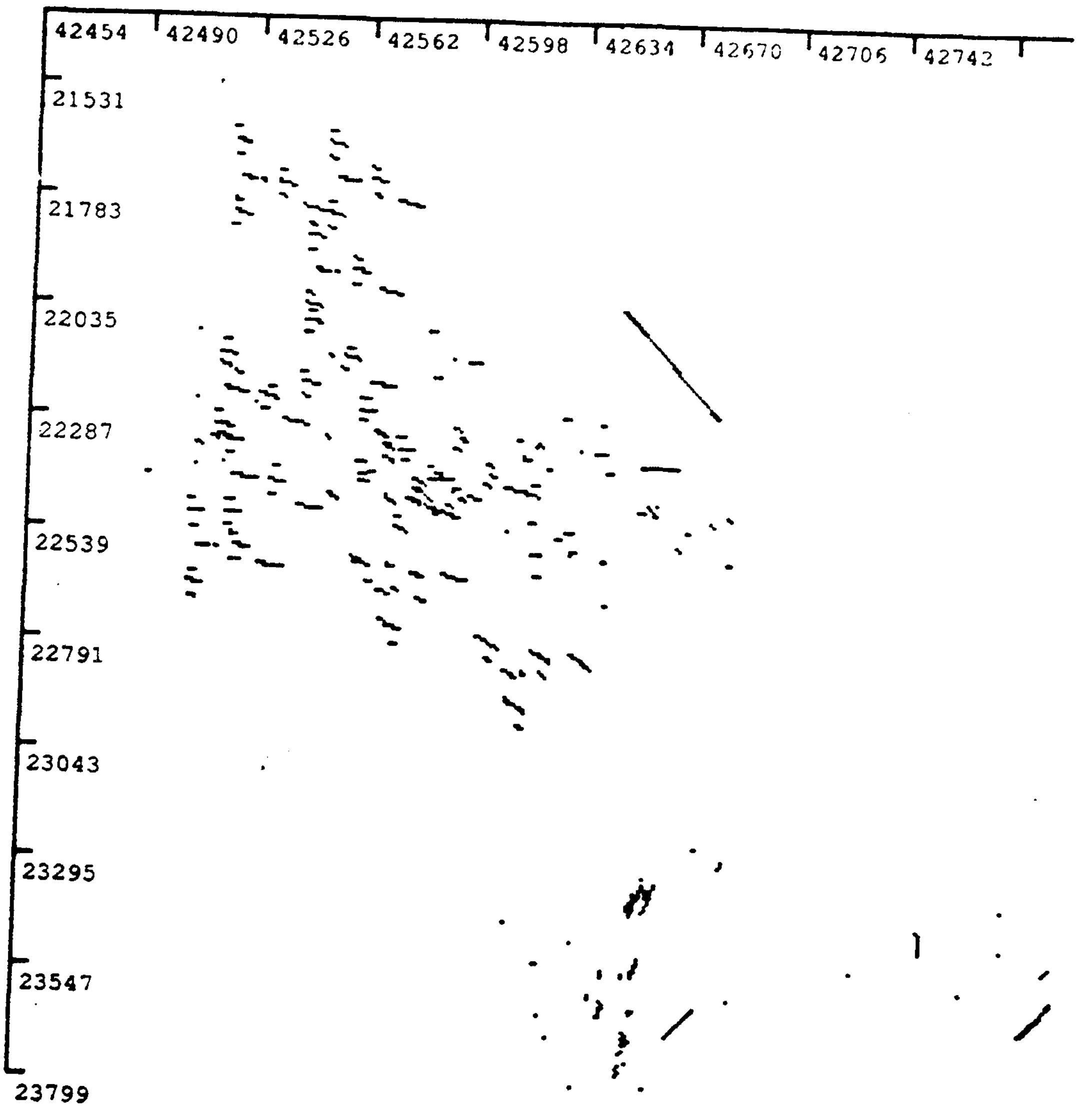

Figure A-178. Input data from data set 56.

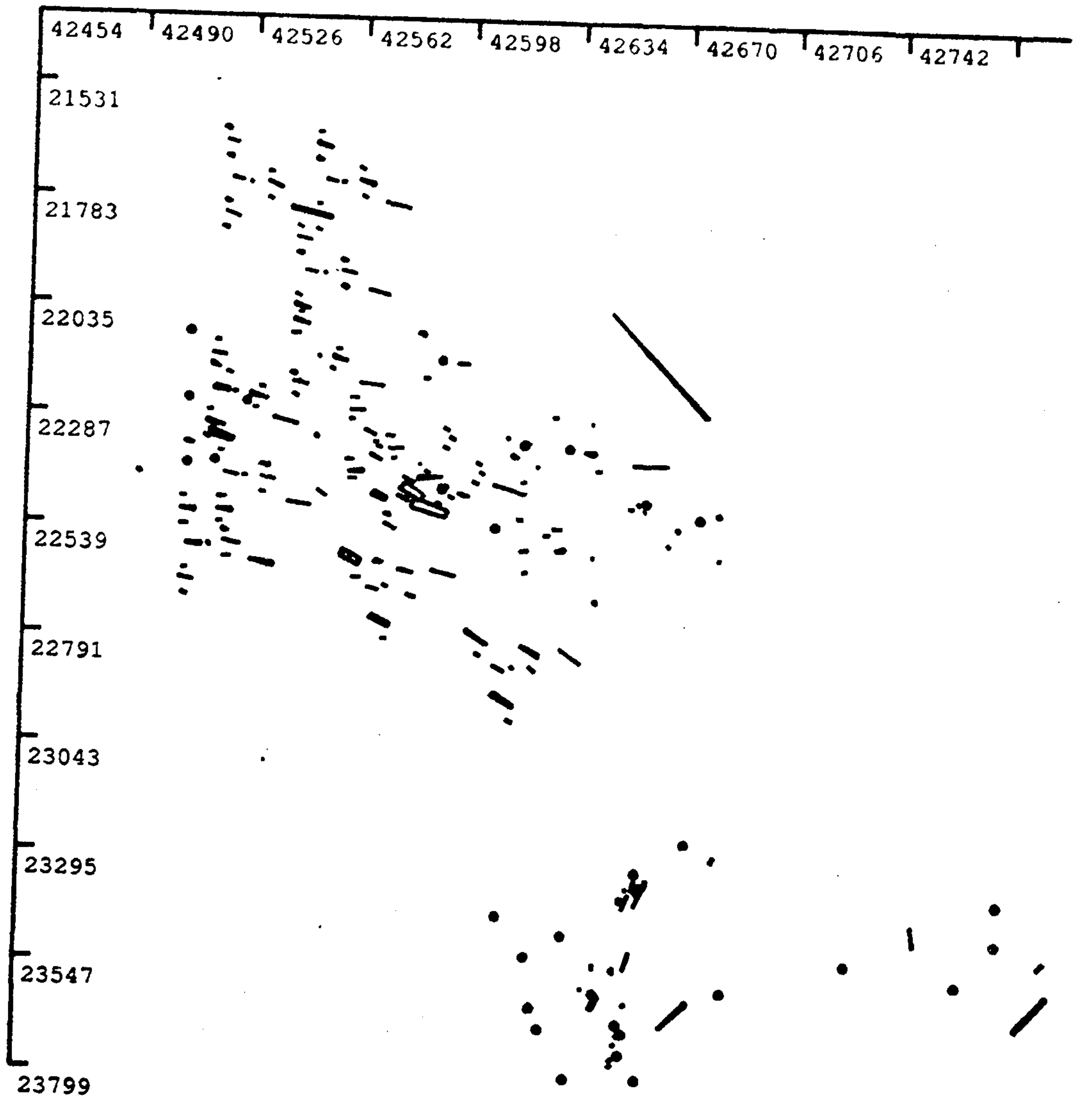

Figure A-179. Initial clusters from data set 56.

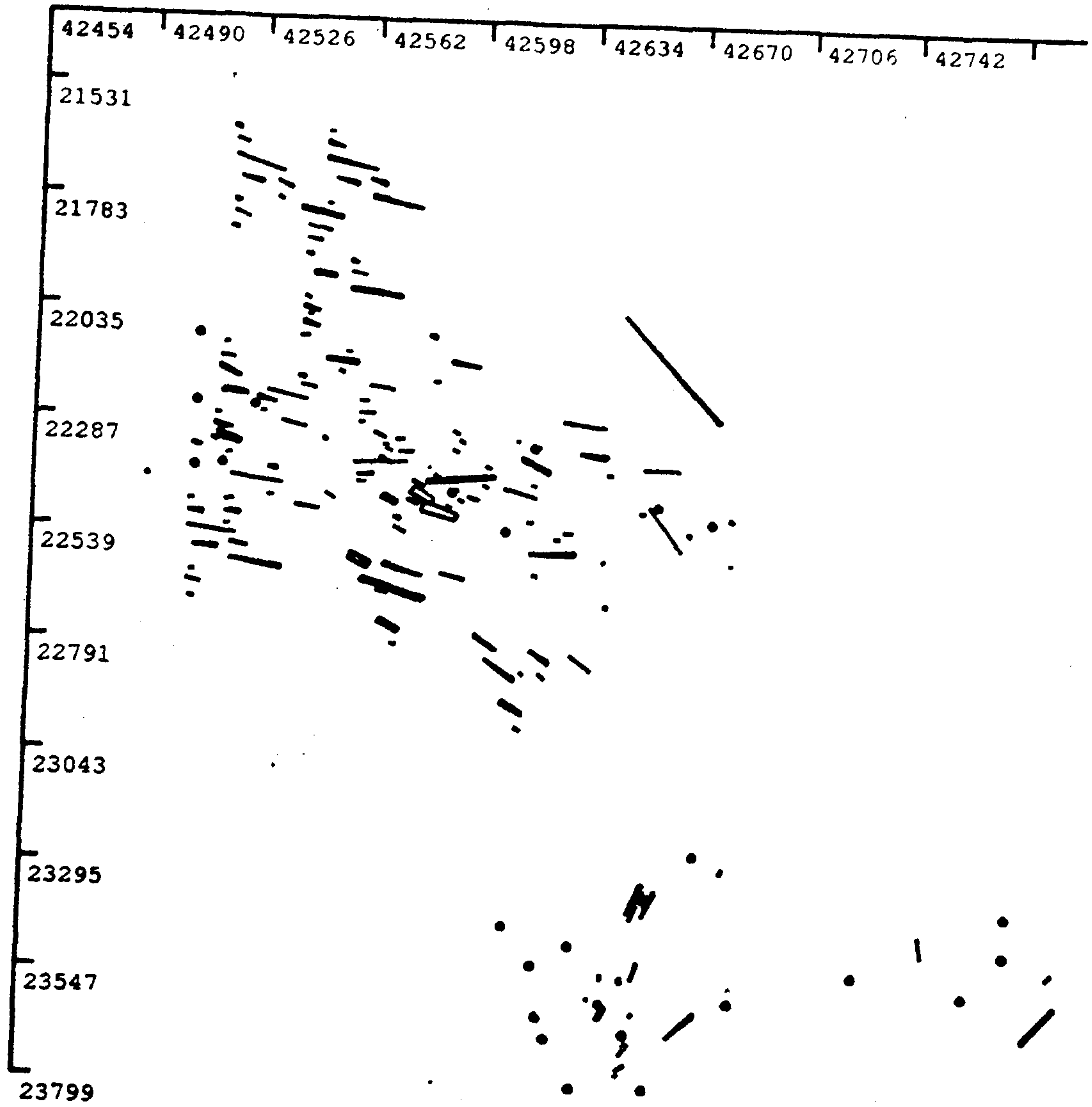

Figure A-180. Linear and online clusters from data set 56.

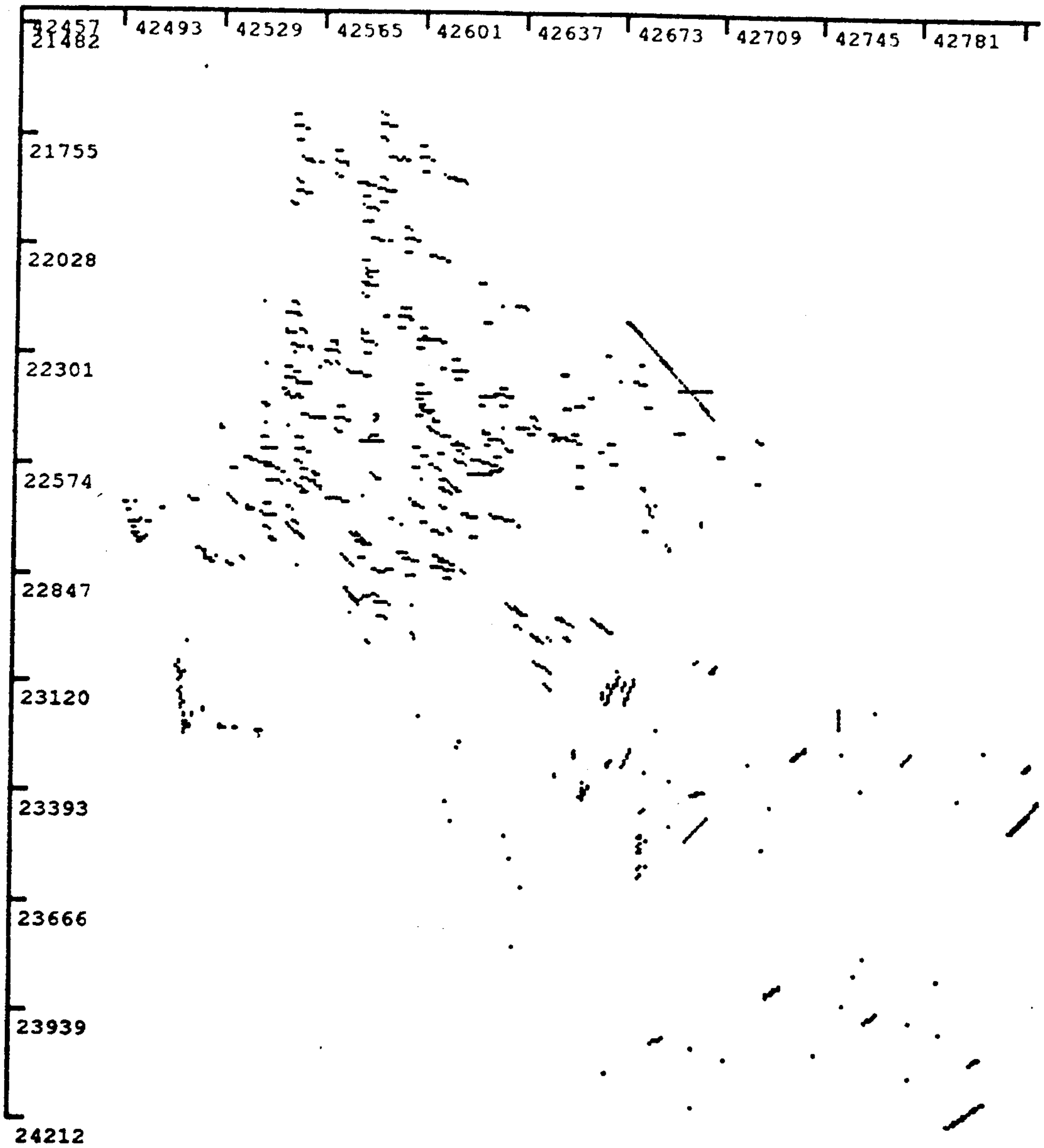

Figure A-181. Input data from data set 57.

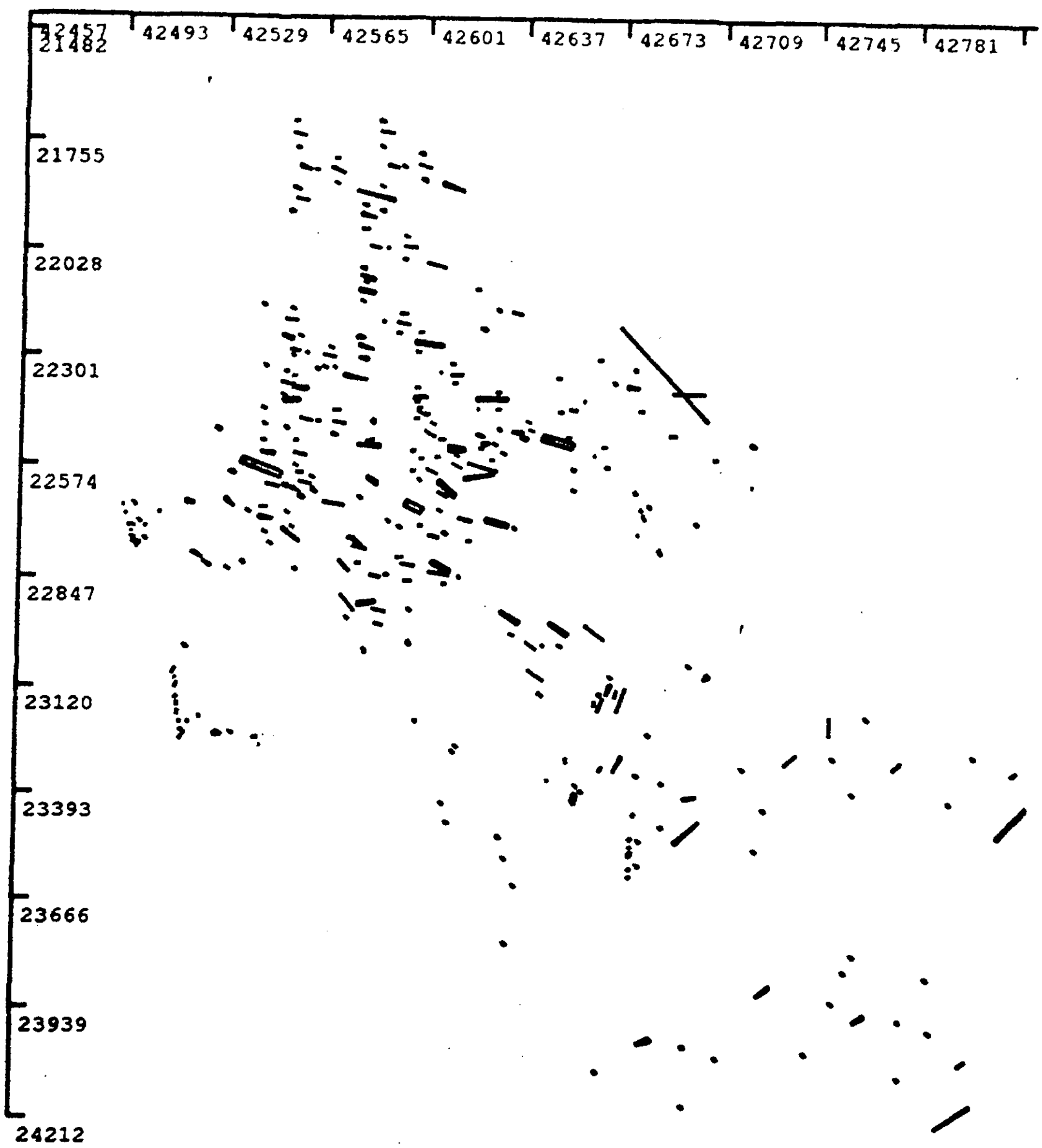

Figure A-182. Initial clusters from data set 57.

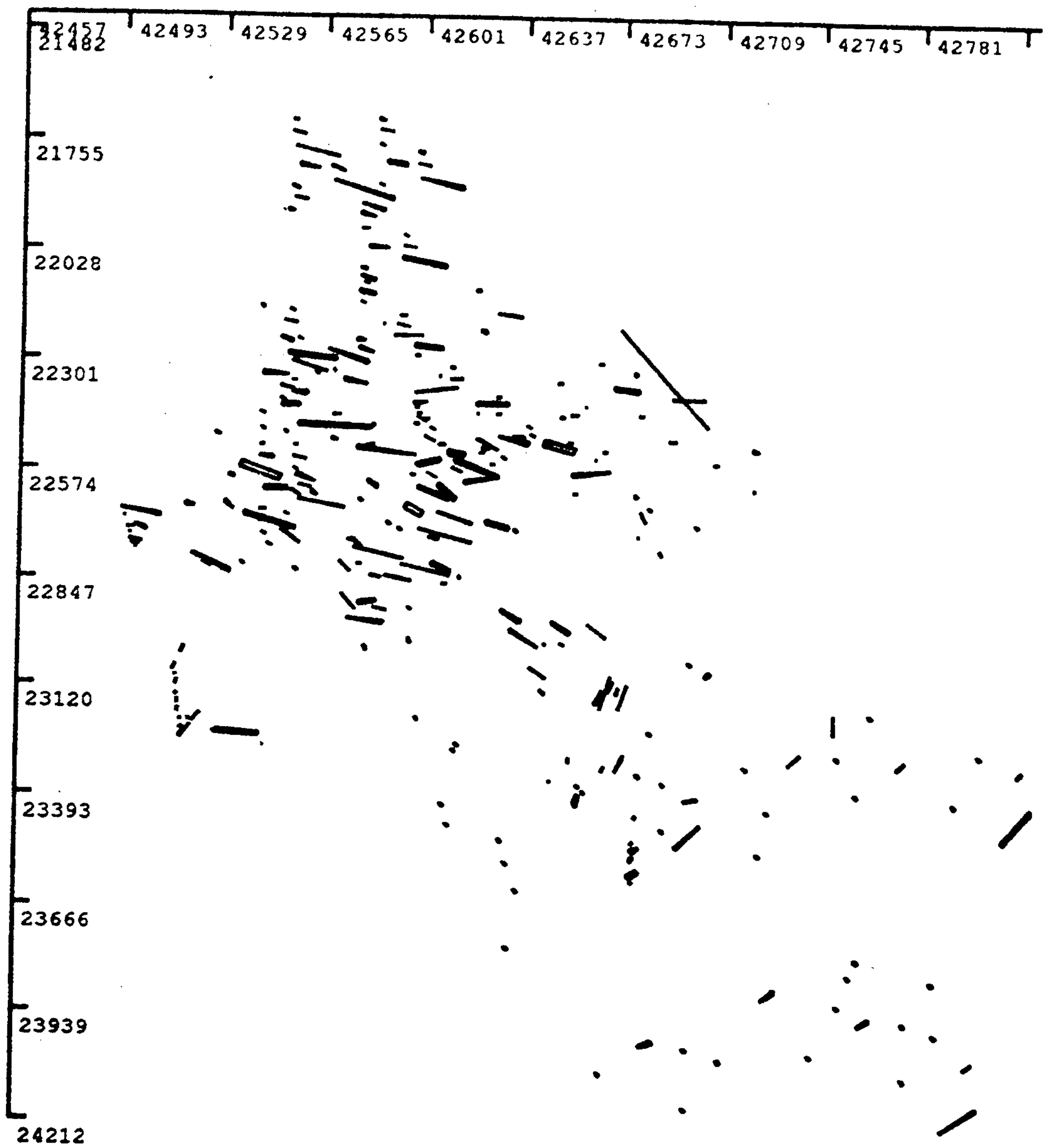

Figure A-183. Linear and online clusters from data set 57.

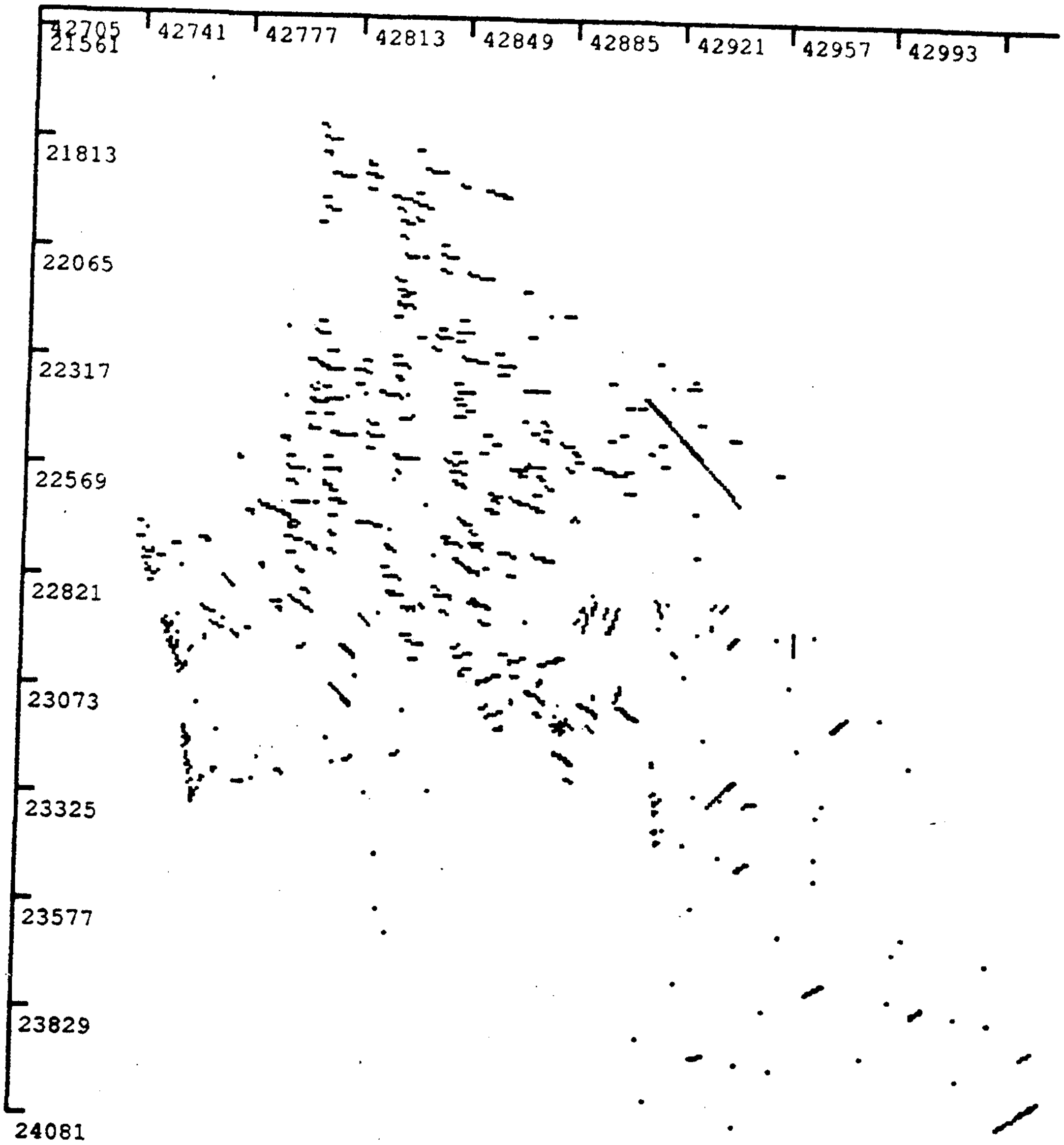

Figure A-184. Input data from data set 58.

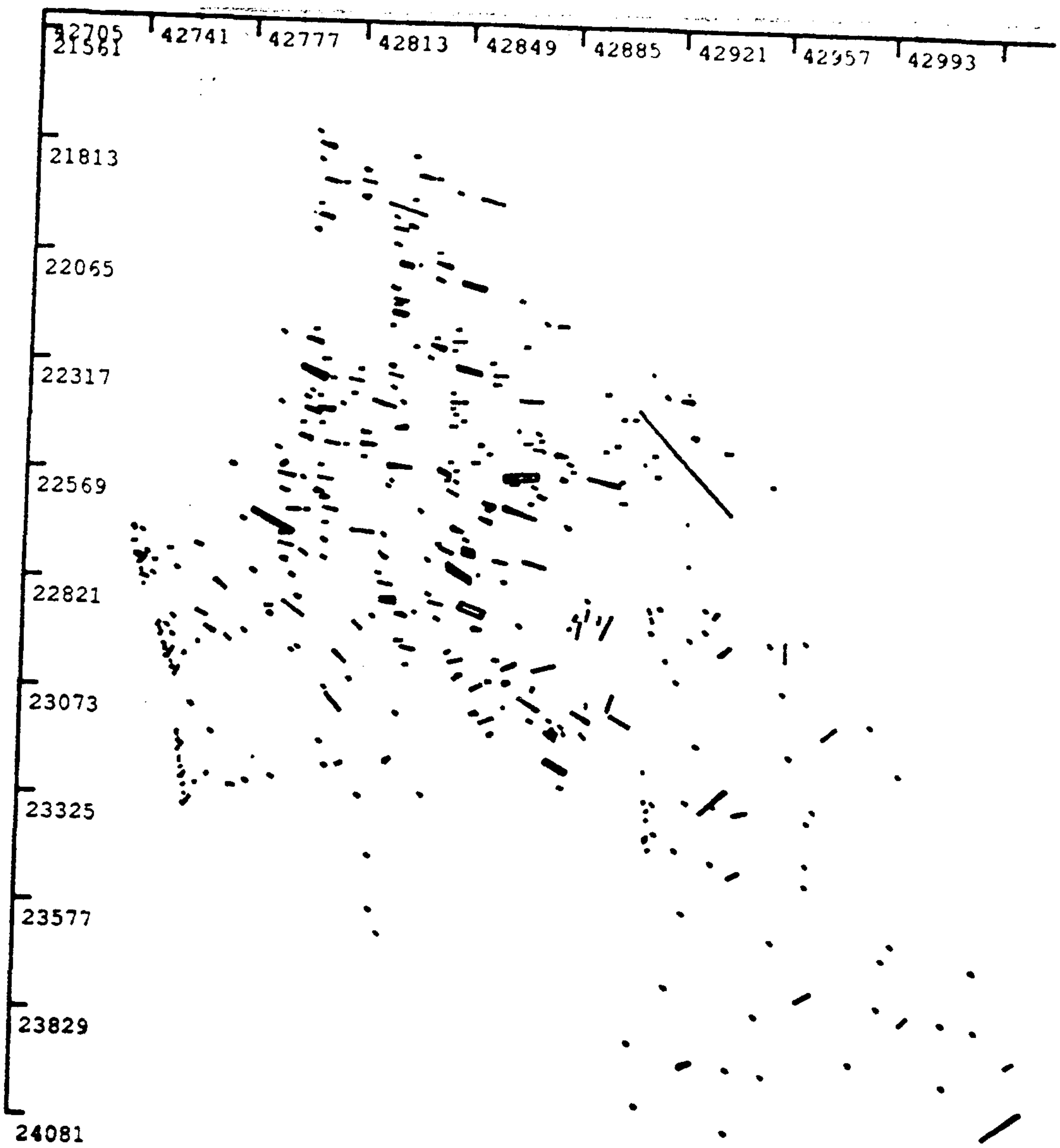

Figure A-185. Initial clusters from data set 58.

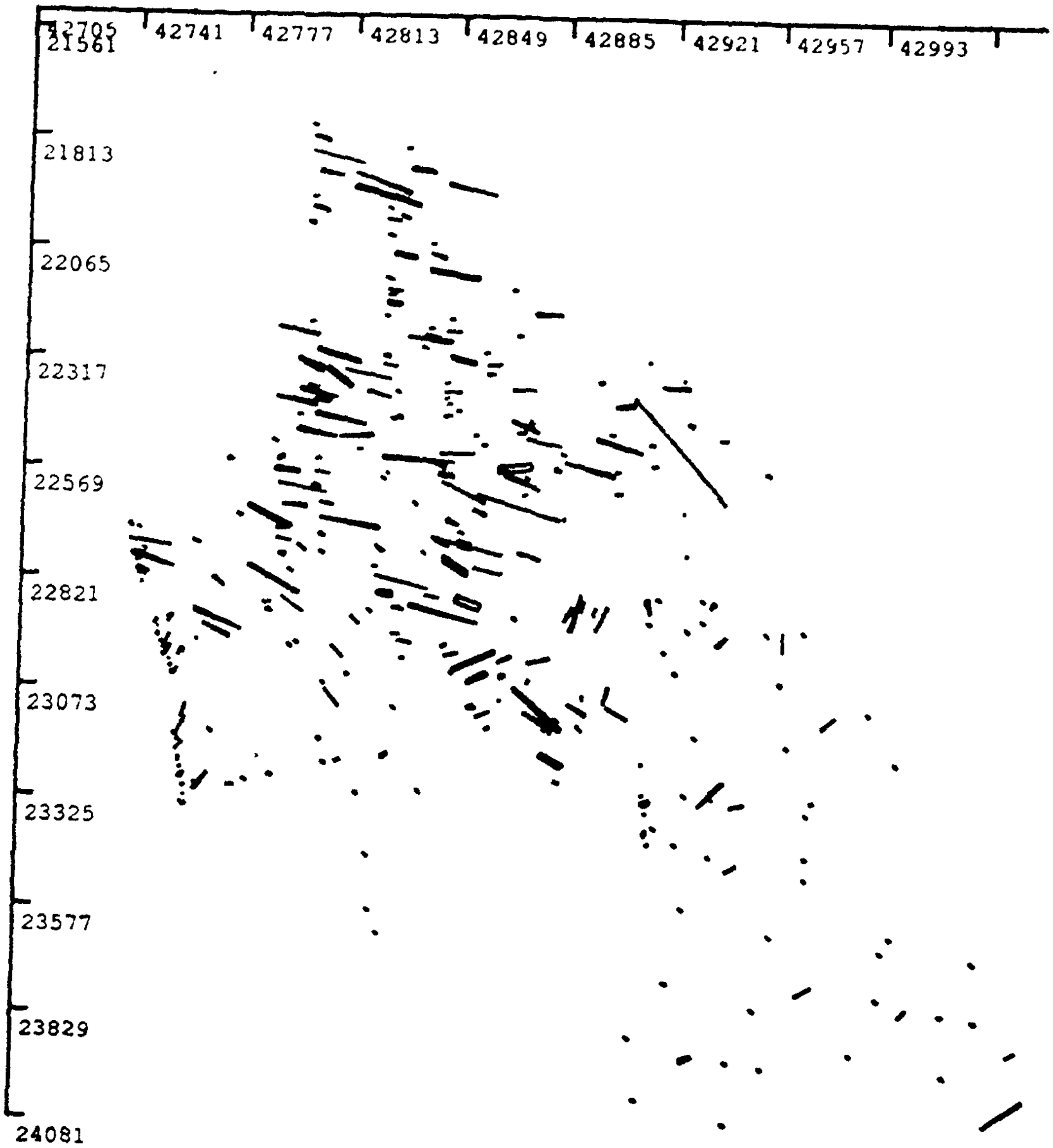

Figure A-186. Linear and online clusters from data set 58.

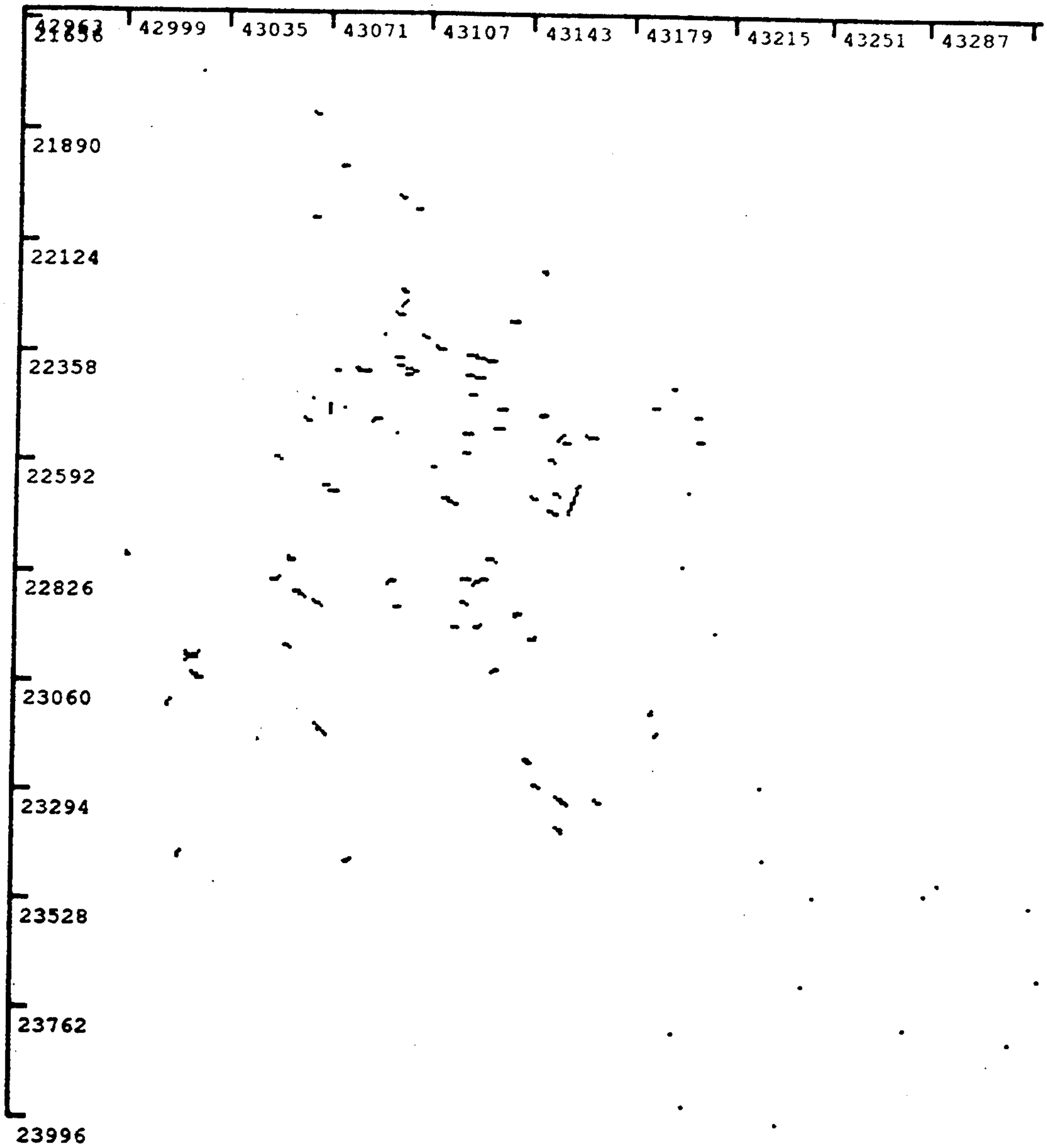

Figure A-187. Input data from data set 59.

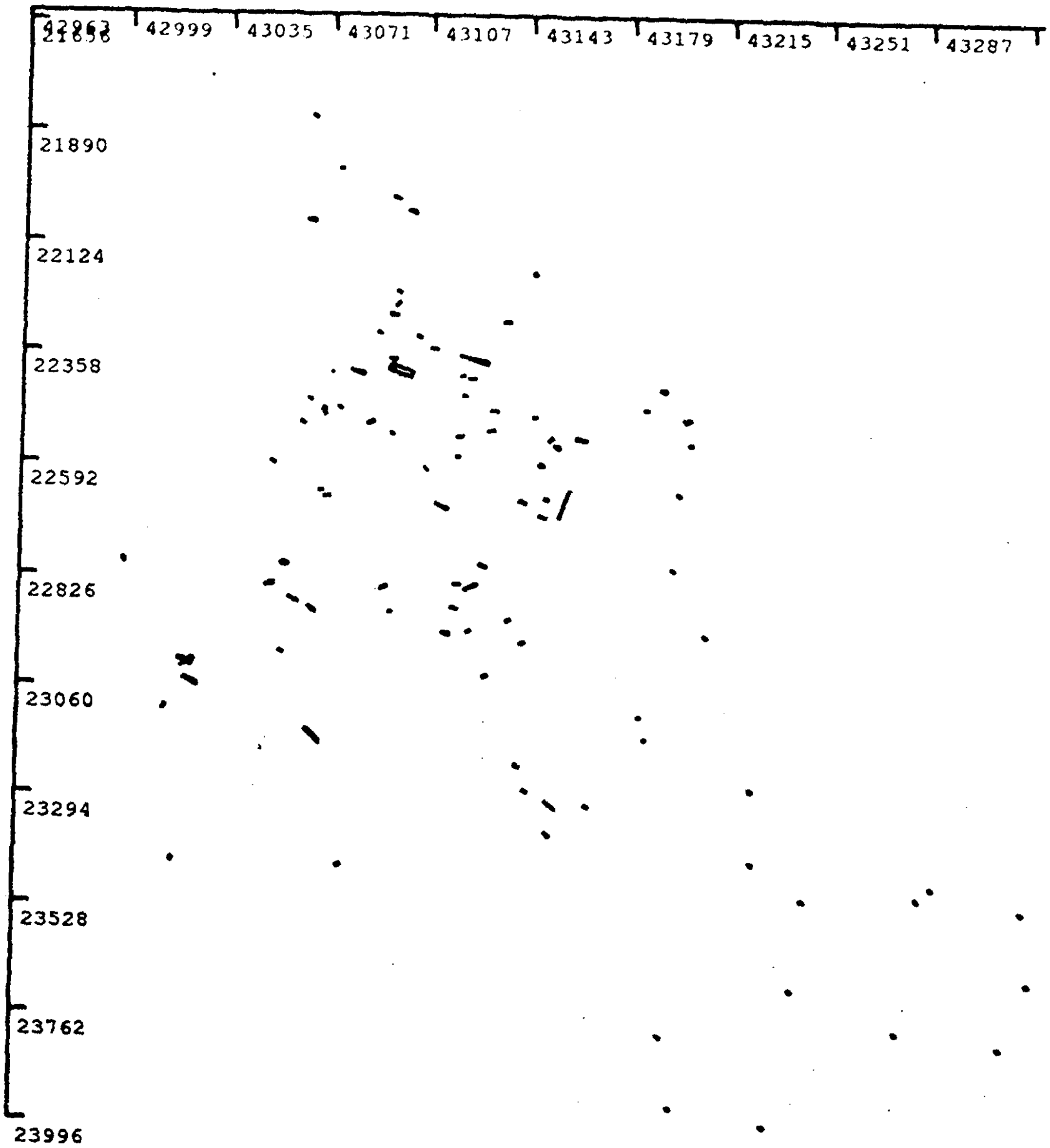

Figure A-188. Initial clusters from data set 59.

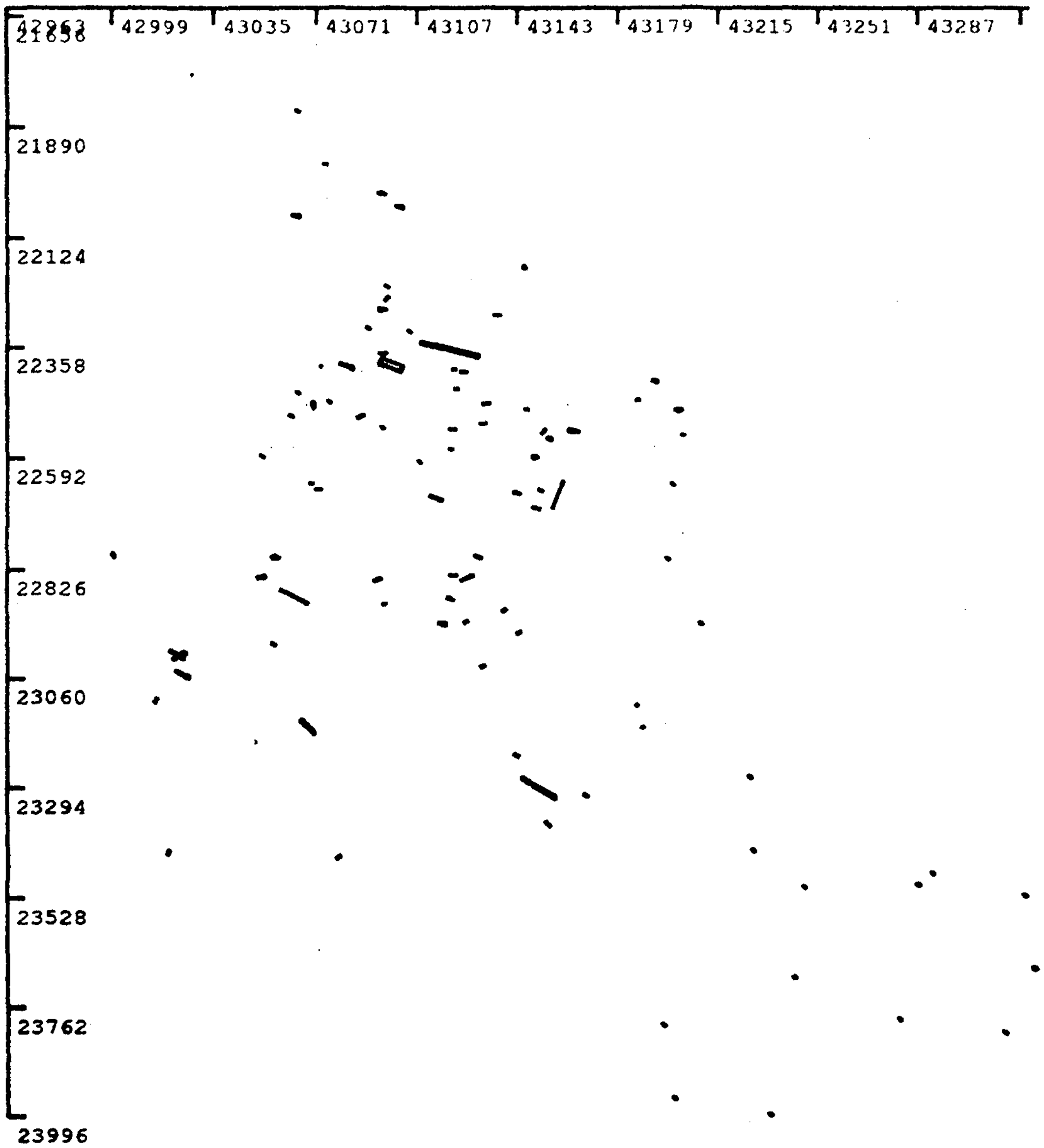

Figure A-189. Linear and online clusters from data set 59.

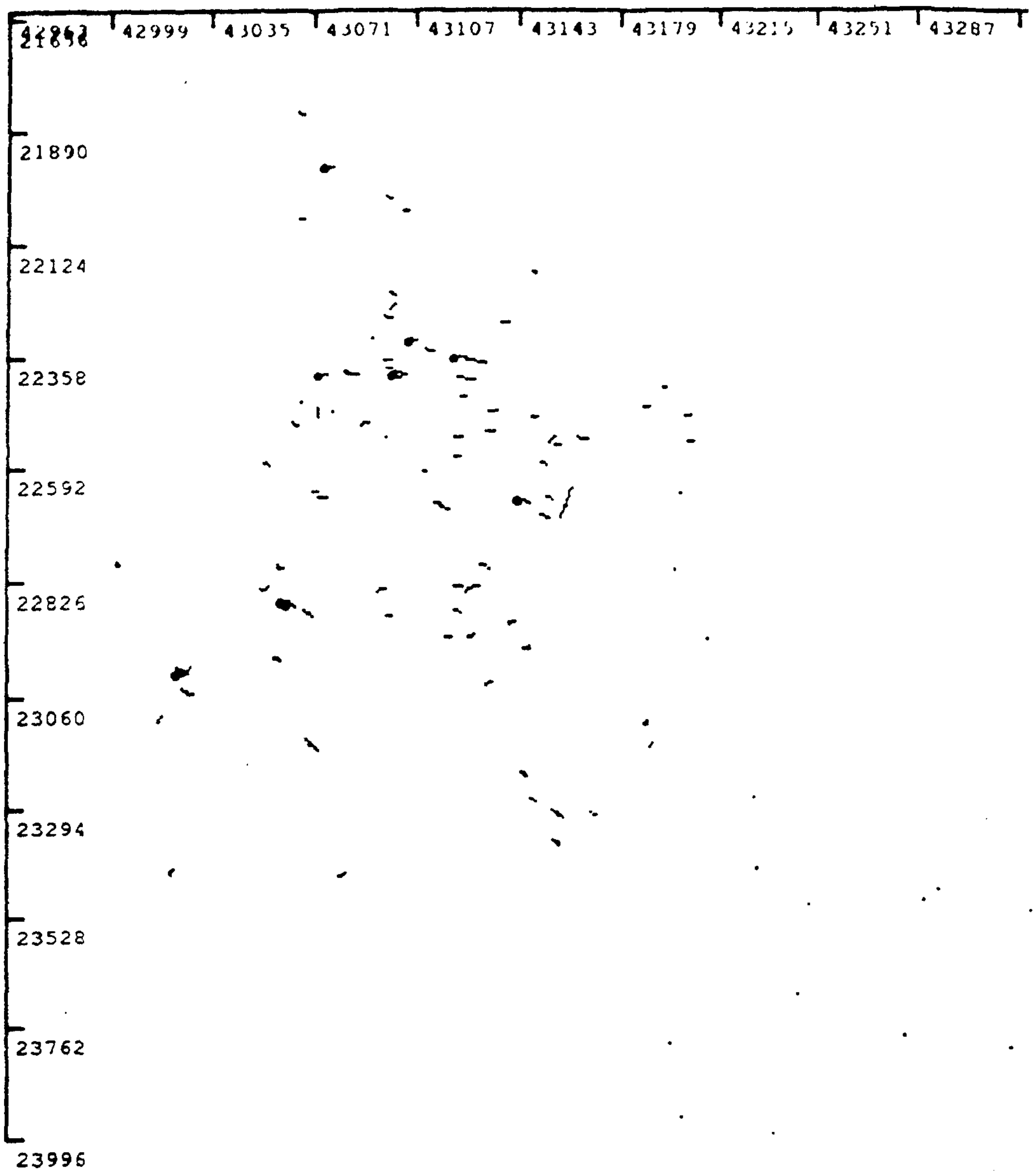

Figure A-190. Circular clustering from data set 59.

NO. OF COPIES	ORGANIZATION
1	DIRECTORATE FOR MANPRINT ATTN HQDA (DAPE MR) DEPUTY CHIEF OF STAFF PERSONNEL 300 ARMY PENTAGON WASHINGTON DC 20310-0300
1	DIRECTOR ARMY AUDIOLOGY & SPEECH CENTER WALTER REED ARMY MEDICAL CENTER WASHINGTON DC 20307-5001
1	OUSD(A)/DDDR&E(R&A)/E&LS PENTAGON ROOM 3D129 WASHINGTON DC 20301-3080
1	CODE 1142PS OFFICE OF NAVAL RESEARCH 800 N QUINCY STREET ARLINGTON VA 22217-5000
1	WALTER REED ARMY INST OF RESEARCH ATTN SGRD UWI C (COL REDMOND) WASHINGTON DC 20307-5100
1	DR ARTHUR RUBIN NATL INST OF STANDARDS & TECH BUILDING 226 ROOM A313 GAITHERSBURG MD 20899
1	COMMANDER US ARMY RESEARCH INSTITUTE ATTN PERI ZT (DR EDGAR M JOHNSON) 5001 EISENHOWER AVENUE ALEXANDRIA VA 22333-5600
1	DEFENSE LOGISTICS STUDIES INFORMATION EXCHANGE US ARMY LOG MGMT COLLEGE FORT LEE VA 23801-6034
1	DEPUTY COMMANDING GENERAL ATTN EXS (Q) MARINE CORPS RD&A COMMAND QUANTICO VA 22134
1	HEADQUARTERS USATRADOC ATTN ATCD SP FORT MONROE VA 23651
1	COMMANDER USATRADOC COMMAND SAFETY OFFICE ATTN ATOS (MR PESSAGNO MR LYNE) FORT MONROE VA 23651-5000

NO. OF COPIES	ORGANIZATION
1	COMMANDER US ARMY MATERIEL COMMAND ATTN AMCAM 5001 EISENHOWER AVENUE ALEXANDRIA VA 22333-0001
1	COMMANDER USA OPERATIONAL T&E AGENCY ATTN CSTE TSM 4501 FORD AVENUE ALEXANDRIA VA 22302-1458
1	USA BIOMEDICAL RSCH & DEV LAB ATTN LIBRARY FORT DETRICK BUILDING 568 FREDERICK MD 21702-5010
1	HQ USAMRDC ATTN SGRD PLC FORT DETRICK MD 21701
1	COMMANDER USA AEROMEDICAL RESEARCH LAB ATTN LIBRARY FORT RUCKER AL 36362-5292
1	US ARMY SAFETY CENTER ATTN CSSC SE FORT RUCKER AL 36362
1	CHIEF ARMY RESEARCH INSTITUTE AVIATION R&D ACTIVITY ATTN PERI IR FORT RUCKER AL 36362-5354
2	DIRECTOR US ARMY RESEARCH LABORATORY ATTN AMSRL OP SD TL (TECH LIB) ADELPHI MD 20783-1145
1	TECHNICAL INFORMATION CENTER HQS TRADOC TEST & EXPERIMENTATION COMMAND EXPERIMENTATION CENTER BLDG 2925 FORT ORD CA 93941-7000
1	US ARMY NATICK RD&E CENTER ATTN STRNC YBA NATICK MA 01760-5020

NO. OF COPIES	ORGANIZATION
1	US ARMY TROOP SUPPORT COMMAND NATICK RD&E CENTER ATTN BEHAVIORAL SCIENCES DIV SSD NATICK MA 01760-5020
1	US ARMY TROOP SUPPORT COMMAND NATICK RESEARCH DEVELOPMENT AND ENGINEERING CENTER ATTN TECH LIBRARY (STRNC MIL) NATICK MA 01760-5040
1	HQ USA RESEARCH INST OF ENVIRONMENTAL MEDICINE ATTN MEDRI CL (DR J KOBRICK) NATICK MA 01760-5007
1	DR RICHARD JOHNSON HEALTH & PERFORMANCE DIVISION US ARIEM NATICK MA 01760-5007
1	LOCKHEED SANDERS INC BOX MER-24-1583 NASHUA NH 03061-0868
1	ATTN DR F WESLEY BAUMGARDNER USAF ARMSTRONG LABORATORY/CFTO SUSTAINED OPERATIONS BRANCH BROOKS AFB TX 78235-5000
1	AFHRL/PRTS BROOKS AFB TX 78235-5601
1	DR JON FALLESEN ARI FIELD UNIT PO BOX 3407 FORT LEAVENWORTH KS 66027-0347
1	COMMANDER USAMC LOGISTICS SUPPORT ACTIVITY ATTN AMXLS AE REDSTONE ARSENAL AL 35898-7466
1	ARI FIELD UNIT FORT KNOX BUILDING 2423 PERI IK FORT KNOX KY 40121-5620
1	COMMANDANT USA ARTILLERY & MISSILE SCHOOL ATTN USAAMS TECH LIBRARY FORT SILL OK 73503

NO. OF COPIES	ORGANIZATION
1	COMMANDER WHITE SANDS MISSILE RANGE ATTN STEWS TE RE WHITE SANDS MISSILE RANGE NM 88002
1	COMMANDER WHITE SANDS MISSILE RANGE ATTN TECHNICAL LIBRARY WHITE SANDS MISSILE RANGE NM 88002
1	USA TRADOC ANALYSIS COMMAND ATTN ATRC WSR (D ANGUIANO) WHITE SANDS MISSILE RANGE NM 88002-5502
1	STRICOM 12350 RESEARCH PARKWAY ORLANDO FL 32826-3276
1	COMMANDER USA TANK-AUTOMOTIVE R&D CENTER ATTN AMSTA RS/D REES WARREN MI 48090
1	COMMANDER USA TANK-AUTOMOTIVE R&D CENTER ATTN AMSTA TSL (TECH LIBRARY) WARREN MI 48397-5000
1	COMMANDER USA COLD REGIONS TEST CENTER ATTN STECR TS A APO AP 96508-7850
1	MR. JOHN HUNT GE BLDG 148-301 ROUTE 38 MOORESTOWN NJ 08057
2	ADMINISTRATOR DEFENSE TECHNICAL INFORMATION CENTER ATTN DTIC DDA 8725 JOHN J KINGMAN RD STE 0944 FT BELVOIR VA 22060-6218
1	US ARMY RSCH DEV STDZN GP-UK ATTN DR MIKE STOUT PSC 802 BOX 15 FPO AE 09499-1500
1	INSTITUTE FOR DEFENSE ANALYSES ATTN DR JESSE ORLANSKY 1801 N BEAUREGARD STREET ALEXANDRIA VA 22311

NO. OF COPIES	ORGANIZATION
1	DR RICHARD W PEW BBN SYSTEMS AND TECHNOLOGY CORP 10 MOULTON STREET CAMBRIDGE MA 02138
1	DR NANCY ANDERSON DEPARTMENT OF PSYCHOLOGY UNIVERSITY OF MARYLAND COLLEGE PARK MD 20742
1	MR LARRY W AVERY BATTELLE PACIFIC NW LABS PO BOX 999 MAIL STOP K6-66 RICHLAND WA 99352
1	LIBRARY ESSEX CORPORATION SUITE 510 1430 SPRING HILL ROAD MCLEAN VA 22102-3000
1	DR BEN B MORGAN DEPARTMENT OF PSYCHOLOGY UNIVERSITY OF CENTRAL FLORIDA PO BOX 25000 ORLANDO FL 32816
1	AFHRL/CA BROOKS AFB TX 78235
1	DR ARTHUR S KAMLET BELL LABORATORIES 6200 EAST BROAD STREET COLUMBUS OH 43213
1	MR AJ ARNOLD STAFF PROJECT ENG HUMAN FACTORS DEPARTMENT GENERAL MOTORS SYSTEMS ENGINEERING 1151 CROOKS ROAD TROY MI 48084
1	DR LLOYD A AVANT DEPARTMENT OF PSYCHOLOGY IOWA STATE UNIVERSITY AMES IA 50010
1	DR PAUL R MCCRIGHT INDUSTRIAL ENGINEERING DEPARTMENT KANSAS STATE UNIVERSITY MANHATTA KS 66502
1	DR MM AYOUB DIRECTOR INSTITUTE FOR ERGONOMICS RESEARCH TEXAS TECH UNIVERSITY LUBBOCK TX 79409

NO. OF COPIES	ORGANIZATION
1	DR TOM MALONE CARLOW ASSOCIATES INC SUITE 750 3141 FAIRVIEW PARK DRIVE FAIRFAX VA 22042
1	DR NORMAN BADLER DEPARTMENT OF COMPUTER AND INFORMATION SCIENCE UNIVERSITY OF PENNSYLVANIA PHILADELPHIA PA 19104-6389
1	COMMANDER US ARMY RESEARCH INSTITUTE OF ENVIRONMENTAL MEDICINE NATICK MA 01760-5007
1	DR DANIEL J POND BATTELLE PNL/K6-66 PO BOX 999 RICHLAND WA 99350
1	HQDA (DAPE-ZXO) ATTN DR FISCHL WASHINGTON DC 20310-0300
1	HUMAN FACTORS ENG PROGRAM DEPT OF BIOMEDICAL ENGINEERING COLLEGE OF ENG & COMPUTER SCIENCE WRIGHT STATE UNIVERSITY DAYTON OH 45435
1	COMMANDER USA MEDICAL R&D COMMAND ATTN SGRD PLC (LTC JJ JAEGAR) FORT DETRICK MD 21701-5012
1	PEO ARMAMENTS ATTN AMCPEO AR BUILDING 171 PICATINNY ARSENAL NJ 07806-5000
1	PEO AIR DEFENSE ATTN SFAE AD S US ARMY MISSILE COMMAND REDSTONE ARSENAL AL 35898-5750
1	JON TATRO HUMAN FACTORS SYSTEM DESIGN BELL HELICOPTER TEXTRON INC PO BOX 482 MAIL STOP 6 FT WORTH TX 76101

NO. OF COPIES	ORGANIZATION
1	DAVID ALDEN HUGHES SIMULATION SYSTEMS INC 5301 E RIVER RD MINNEAPOLIS MN 55421-1024
1	OASD (FM&P) WASHINGTON DC 20301-4000
1	COMMANDER US ARMY MATERIEL COMMAND ATTN AMCDE AQ 5001 EISENHOWER AVENUE ALEXANDRIA VA 22333
1	COMMANDER MARINE CORPS SYSTEMS COMMAND ATTN CBGT QUANTICO VA 22134-5080
1	DIRECTOR AMC-FIELD ASSISTANCE IN SCIENCE & TECHNOLOGY ATTN AMC-FAST (RICHARD FRANSEEN) FT BELVOIR VA 22060-5606
1	COMMANDER US ARMY FORCES COMMAND ATTN FCDJ SA BLDG 600 AMC FAST SCIENCE ADVISER FT MCPHERSON GA 30330-6000
1	COMMANDER I CORPS AND FORT LEWIS AMC FAST SCIENCE ADVISER ATTN AFZH CSS FORT LEWIS WA 98433-5000
1	HQ III CORPS & FORT HOOD OFFICE OF THE SCIENCE ADVISER ATTN AFZF CS SA FORT HOOD TX 76544-5056
1	COMMANDER U.S. ARMY NATIONAL TRAINING CENTER AMC FAST SCIENCE ADVISER ATTN AMXLA SA FORT IRWIN CA 92310
1	COMMANDER HQ XVIII ABN CORPS & FORT BRAGG OFFICE OF THE SCI ADV BLDG 1-1621 ATTN AFZA GD FAST FORT BRAGG NC 28307-5000

NO. OF COPIES	ORGANIZATION
1	SOUTHCOM WASHINGTON FIELD OFC 1919 SOUTH EADS ST SUITE L09 AMC FAST SCIENCE ADVISER ARLINGTON VA 22202
1	HQ US SPECIAL OPERATIONS COMMAND AMC FAST SCIENCE ADVISER ATTN SOSD MACDILL AIR FORCE BASE TAMPA FL 33608-0442
1	HQ US ARMY EUROPE AND 7TH ARMY ATTN AEAGX SA OFFICE OF THE SCIENCE ADVISER APO AE 09014
1	COMMANDER HQ 21ST THEATER ARMY AREA COMMAND AMC FAST SCIENCE ADVISER ATTN AERSA APO AE 09263
1	COMMANDER HEADQUARTERS USEUCOM AMC FAST SCIENCE ADVISER UNIT 30400 BOX 138 APO AE 09128
1	HQ V CORPS COMMAND GROUP UNIT #25202 AMC FAST SCIENCE ADVISER ATTN AETV SA APO AE 09079-0700
1	HQ 7TH ARMY TRAINING COMMAND UNIT #28130 AMC FAST SCIENCE ADVISER ATTN AETT SA APO AE 09114
1	COMMANDER HHC SOUTHERN EUROPEAN TASK FORCE ATTN AESE SA BUILDING 98 AMC FAST SCIENCE ADVISER APO AE 09630
1	COMMANDER US ARMY PACIFIC AMC FAST SCIENCE ADVISER ATTN APSA FT SHAFTER HI 96858-5L00

NO. OF COPIES	ORGANIZATION
1	COMMANDER US ARMY JAPAN/IX CORPS UNIT 45005 ATTN APAJ SA AMC FAST SCIENCE ADVISERS APO AP 96343-0054
1	AMC FAST SCIENCE ADVISERS PCS #303 BOX 45 CS-SO APO AP 96204-0045
1	COMMANDER ALASKAN COMMAND ATTN SCIENCE ADVISOR (MR GRILLS) 6-900 9TH ST STE 110 ELMENDORF AFB ALASKA 99506
1	CDR & DIR USAE WATERWAYS EXPERIMENTAL STA ATTN CEWES IM MI R (AS CLARK CD DEPT #1153) 3909 HALLS FERRY ROAD VICKSBURG MS 39180-6199
1	DIRECTOR US ARMY RESEARCH LABORATORY ATTN AMSRL OP SD TP (TECH PUB) ADELPHI MD 20783-1145
1	DIRECTOR US ARMY RESEARCH LABORATORY ATTN AMSRL OP SD TA (REC MGMT) ADELPHI MD 20783-1145
1	DR SEHCHANG HAH DEPT OF BEHAVIORAL SCIENCES & LEADERSHIP BUILDING 601 ROOM 281 US MILITARY ACADEMY WEST POINT NEW YORK 10996-1784
1	US ARMY RESEARCH INSTITUTE ATTN PERI IK (DOROTHY FINLEY) 2423 MORANDE STREET FORT KNOX KY 40121-5620
5	CHIEF ARL HRED USAFAS FIELD ELEMENT ATTN AMSRL HR MF (L PIERCE) BLDG 3040 ROOM 220 FORT SILL OK 73503-5600
5	DEBORAH TRYTTEN SCHOOL OF COMPUTER SCIENCE UNIVERSITY OF OKLAHOMA NORMAN OK 73069

NO. OF COPIES	ORGANIZATION
	ABERDEEN PROVING GROUND
5	US ARMY RESEARCH LABORATORY ATTN AMSRL OP AP L (TECH LIB) BLDG 305
1	ARL LIBRARY BLDG 459
1	ARL SLAD ATTN AMSRL BS (DR JT KLOPCIC) BLDG 328 APG-AA
1	COMMANDER CHEMICAL BIOLOGICAL AND DEFENSE COMMAND ATTN AMSCB CI APG-EA
1	USATECOM RYAN BUILDING APG-AA